AFRL-RX-WP-TR-2012-0345

HUMAN-ASSISTED-MANUFACTURING MODEL LIBRARY

Jonathan Vance, Douglas A. Stuart, and Steven A. Dorris
Boeing Company

Frank Liou
Missouri S&T

Robert Tilove
General Motors

Gabor Karsai
Vanderbilt

JUNE 2012
Final Report

STINFO COPY

AIR FORCE RESEARCH LABORATORY
MATERIALS AND MANUFACTURING DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7750
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

_____**//SIGNED//**_____   _____**//SIGNED//**_____
C. BRANDON LOVETT, Program Manager   SCOTT M. PEARL, Branch Chief
AFRL/RXMS   AFRL/RXMS

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*	
June 2012	Final	2 May 2011 – 6 June 2012	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
HUMAN-ASSISTED-MANUFACTURING MODEL LIBRARY	FA8650-11-2-7127
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 62303E

6. AUTHOR(S)	5d. PROJECT NUMBER
Jonathan B. Vance , Douglas A. Stuart, and Steven A. Dorris (Boeing)	3000
Frank Liou (Missouri S&T),	**5e. TASK NUMBER** 00
Robert Tilove (General Motors), Gabor Karsai (Vanderbilt)	**5f. WORK UNIT NUMBER** M0129800

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Boeing Company Boeing Research and Technology P.O. Box 516 St. Louis, MO 63166-0516	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY ACRONYM(S)
Air Force Research Laboratory Materials and Manufacturing Directorate Wright-Patterson Air Force Base, OH 45433-7750 Air Force Materiel Command United States Air Force	DARPA/TTO 675 North Randolph Street Arlington, VA 22203-2114	AFRL/RXMS
		11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-RX-WP-TR-2012-0345

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
The U.S. Government is joint author of this work and has the right to use, modify, reproduce, release, perform, display, or disclose the work. PA Case Number and clearance date: 88ABW-2012-3763, 5 July 2012. This document contains color.

14. ABSTRACT
Manufacturing process models are developed for a data-driven, reconfigurable manufacturing facility. A human assembly process model is developed to characterize assembly processes. Machine and tool resources are characterized for manufacturing processes to provide a broad characterization of manufacturing process coverage in a hypothetical final assembly factory for infantry fighting vehicles. A sample population of manufacturing process parameters and resources is collected and incorporated into a manufacturing capability and prcess model library. Requirements for a virtual manufacturing environment are developed, and an initial, prototype version is implemented to facilitate process model validation, configuration, and experimentation.

15. SUBJECT TERMS
Assembly, Manufacturing, Process Model, Adaptive Vehicle Make, iFAB, Manufacturing Model Library, AVM, MCPML, Serious Game, VME, Human

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON *(Monitor)*
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	238	C. Brandon Lovett **19b. TELEPHONE NUMBER** *(Include Area Code)* N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

TABLE OF CONTENTS

Section   **Page**

LIST OF FIGURES

Figure   **Page**

Figure **Page**

v

LIST OF TABLES

Table   **Page**

1.0   SUMMARY

The instant Foundry Adaptive through Bits (iFAB) relies on a manufacturing model library as the datastore for process and capability information. This report discusses the implementation of a manufacturing library and its ancillary components.

A manufacturing model language is created by characterizing aspects of a manufacturing facility alternatively referred to as a foundry. The foundry aspects are identified as machines, tools, material handling equipment, and other tangible resources. Infrastructure and facilities details are characterized. The human actor of manufacturing is captured with respect to actions and effects the human actor has on fabrication and assembly activities. All of these foundry features and elements are semantically defined in a manufacturing capability and process model language.

The aforementioned manufacturing language defines a library with data about machines, tools, processes, resources, human activities, aggregated process descriptions, and more. This Manufacturing Capability and Process Model Library (MCPML) is presented to external tools via an interface. That interface to the MCPML contains the logic and heuristics needed to place the manufacturing elements and resources into appropriate context for manufacturing processes. The MCPML interface provides data to queries for manufacturability, process sequencing, and foundry configuration. The interface is extensible and service-oriented.

A prototype Virtual Manufacturing Environment (VME) is created to ascertain manufacturability from a crowd-sourced perspective and virtual configuration of the iFAB foundry. VME is created in an open-source virtual worlds engine. Assembly process information is pulled from the MCPML interface as XML data which shows, through use of avatars, manufacturing feedback on reachability, spatial configuration, part and work cell layout, and potential for collaboration between multiple users.

2.0   INTRODUCTION

This report describes the Human-Assisted-Manufacturing Model Library (HAMML) developed by Boeing for the DARPA manufacturing model library in iFAB, which is an Adaptive Vehicle Make (AVM) program.   HAMML includes a library of various fabrication processes and associated factory components, such as machines and techniques to produce the elements and assemble the product. The principal thrust of this technical effort is characterization and definition of human modeling that is pertinent to manufacturing such as manual task times, anthropometric force limits, and accessibility constraints. Another key element of this effort is the characterization of various manufacturing machines, techniques, and processes in terms of their range of applicability, range of reconfigurability, cost, speed, and other relevant characteristics. It supports development of a coherent manufacturing library through creation of a language that enables the uniform representation of various library components and their models.

This supports the MCPML component of the iFAB project which provides for rapid design and configuration of manufacturing capabilities, initially to support the fabrication of military ground vehicles and variants. The iFAB end vision is that of a facility which can fabricate and assemble responsive designs, verified and supplied in a comprehensive meta-language representation.

iFAB includes existing fabrication capabilities described by a model library that characterizes the salient attributes of each modality of fabrication in terms including: cost, speed, range of applicability, and speed of re-configurability. The resultant factory or foundry can range in form from a single facility under one roof to a virtual aggregation of distributed capabilities, sequenced and tied together into a single resultant product flow.
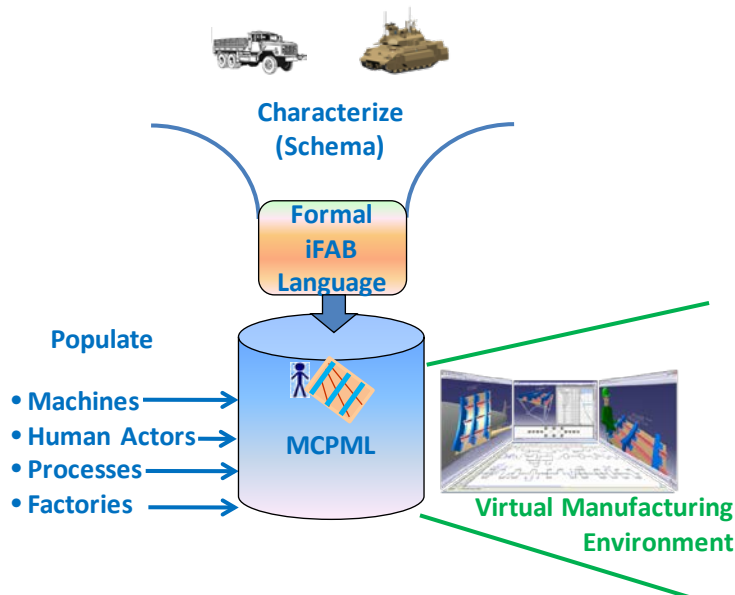
Figure 1. Overview of MCPML Approach

The HAMML activity is organized into a set of technical and management tasks as specified by the contract Statement of Work and summarized here.

Task One characterized the manufacturing machine and tool space by identifying representative machines and tools to support the manufacturing and assembly of an armed, armored and manned military ground vehicle, primarily in manufacturability terms such as size, weight, materials, tolerances, and access. This task also describes the suite of machines for creating and finishing individual component parts and sections, and the assembly processes required to combine pieces into systems and subsystems, including final assembly of the completed vehicle. This task has involved researching, with Missouri University of Science and Technology (MS&T), the machines and tools used to construct Infantry Fighting Vehicle (IFV) system components to define machine characteristics and refine the scope for population of the MCPML. This included complex motion. The task also has involved analyzing machines and tools used in aerospace and automotive industries for relevance to combat vehicle with emphasis on extracting machine and tool characteristics (initially complex CNC machines) used in a similar domain. This task has involved consulting with machine developers in support of scoping the machine and tool space and defining characteristics to include in the MCPML schema.

Task Two characterized the factory, manufacturing/assembly processes, and human actor space required to support production and assembly of vehicles and determined the information required to fully model a factory for the MCPML. Plant infrastructure includes floor space, power/infrastructure requirements/availability, workstations, as well as re-configurability parameters including time to reconfigure space, work cells, and move components. This task process space includes needed manufacturing and assembly processes and machines and tools. The human actor space describes the required operator actions, operator skill sets, and robotic support needed to perform the manufacturing and assembly tasks and was determined through research concerning factories spaces, manufacturing / assembly processes, and human actors to define principal characteristics and refine the scope of the space for population in MCPML. The task also built and reviewed/refined with factory experts a candidate factory schema that was then compared with facilities for existing ground combat vehicle facilities. As with other tasks, this development included analyzing processes in relevant automotive and aerospace manufacturing examples to extract features that may be common to the IFV domain. The information gathered enabled formulation of a schema for MCPML factories, processes, and humans.

Task Three defined the language for the MCPML by capturing and expressing the attributes and characteristics of manufacturing capabilities and processes identified in task one. The language captures machine characteristics (size, weight, power, accuracy, variances, cost, speed, range of applicability, speed of reconfigurability); machine capabilities (materials, production process, part size and weight); and capabilities required of, and constraints resulting from, human process participants (ergonomic, range of motion, accessibility, force required, etc.). This task was strongly driven by the work in tasks one, two and four (described below) in that those are the tasks that determined fundamental characteristics of machines, tools, processes, and human actor along with their virtual representations. The task developed syntax and semantics required to capture the characteristics, constructed metamodels for the language, and generated the language and library implementation.

Task Four defined a Virtual Manufacturing Environment as a tool to facilitate manufacturing modeling and foundry configuration. VME is based on a "serious game" model including human actors completing processes for manufacturing selected components. This task also developed a prototype VME to MCPML interface definition and requirements for the virtual environment. For this task, serious-game frameworks were evaluated and one was selected to serve as the basis for MCPML VME component. The tasks also defined the MCPML virtual environment and corresponding schema and constructed an initial prototype. This prototype is exercised with scenarios drawn from IFV challenge problems.

Task Five captured and populated the MCPML library with capability and process definitions in the MCPML language for a representative cross section of machines and process identified in Task Two required to support iFAB creating and assembling a representative cross section of the vehicle components identified in Task One. This included identifying machines, tools, processes used to construct the initial library and collection of detailed characteristics concerning selected elements for library population. The task created MCPML entries with appropriate factories and human element models employing selection criteria including coverage of a significant and representative slice of IFV domain and supporting the experimentation activities in Task Six.

Task Six involved experiments, demonstrations of MCPML capabilities, and integration activities with the customer and other iFAB, META, and AVM recipients. This demonstrated the MCPML interface tool as part of the greater AVM tool-chain and provided internal validation of the language and library. This integration ensured compatibility of the language and library with the broader iFAB, META and AVM programs and supported program-wide integration experiments. This task included developing a detailed experimentation plan that ensured experiments and data collection showed the enhanced effectiveness of iFAB compared with current approaches. This plan also identified and defined challenge problem vignettes for incremental experimentation with MCPML. Some of the experiments involved generating configurations without the benefit of TA 1 algorithms and investigating creating algorithm stubs or emulations of TA 1 algorithms.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

The overall procedure for this research effort begins with characterization of the manufacturing facility from the perspective of defining an ontological representation that will facilitate modeling of manufacturing capabilities and processes in a software tool chain. Then, having defined the language for MCPML, population of the library and implementation of the various interfaces to other tools is executed. Among interfaces to iFAB foundry configuration, process planning, and manufacturability tools, the MCPML is interfaced to the VME.

3.1 Define Assembly Process & Human Factor Space

To develop a taxonomy based on automotive assembly tasks, multiple sources were used to assess and characterize the capabilities of a human operator.

Sufficient characterization of human assembly capabilities required analysis of multiple assembly task assessment sources and development a human characterization assembly taxonomy. Analysis of the previously described General Motors operator instructions provided a starting point for task assessment characterization. The assessment continued by evaluating the Boeing human operations schema obtained from the Industrial Engineering department. Finally, General Motors standard time assessment and MODAPTS were also examined to contribute to the human capabilities assembly taxonomy. Human and hand posture libraries were generated along with associated whole body peak strength values. Plant equipment, material and tool characterization was also developed to associate tasks with the appropriate plant elements like equipment and tools.

3.1.1 Analysis Work Characteristics

Work descriptions were analyzed to better understand underlying work elements needed to model operations within a plant. For each operation within the example vehicles task elements were identified including the number of people required to perform the operation, the requirement for a moving assembly line, the requirement to manipulate flexible parts and rigid parts, requirements to install fasteners, and the need to use hand tools and fixtures or lift assists. Tables 1and 2 document the resulting analysis.

All tasks were performed on a moving assembly line. One to three operators were needed to perform each operation. Most of the operations studied included flexible parts like electrical cables and hoses as well as rigid parts. Most required the installation of fasteners like bolts, screws and clamps. Hand tools are required to perform most operations studied, however only a few work cases required lift assists or fixtures to perform the required tasks.

Table 1. Truck Operation Analysis

Operation Instruction Sheets	# People	Moving Line	Flexible Parts	Rigid Parts	Fasteners	Hand Tools	Fixtures/Lift Assists
IP Installation							
IP Assist	3	Y	Electrical	Y	Bolts (2)	Right Angle Tool	N
IP Load		Y	Electrical	Y	Bolts (2)		IP Load Tool
IP Secure to Dash		Y	Electrical	Y	Bolts (4)		N
Engine Marriage							
Engine Assist	2	Y	Power Steering Lines	Y	N	N	N
Set Engine		Y	Electrical	Y	N	N	Engine Bridle
Secure Right Engine Mount	1	Y	N	Y	Bolts (3)	Right Angle Tool	N
Secure Left Engine Mount	1	Y	N	Y	Bolts (3)		N
Engine Dress/Heater Hose							
Heater Hose	1	Y	Hose	N	N	N	N
Engine Dress/BEC Module							
BEC Tray Install	1	Y	N	Y	Bolts (3)	Right Angle Tool	N
Secure Electrical Connections	1	Y	Electrical	Y	N	N	N

Table 2. Car Operation Analysis

Operation Instruction Sheets	# People	Moving Line	Flexible Parts	Rigid Parts	Fasteners	Hand Tools	Fixtures/Lift Assists
IP Installation							
IP Load	3	Y	Electrical	Y	N	N	N
Left IP Assist and Secure		Y	Electrical	Y	Bolts (3)	Right Angle Tool	N
Right IP Assist and Secure		Y	Electrical	Y	Bolts (3)	Right Angle Tool	IP Load Tool
Engine Marriage							
Set Engine	2	Y	Electrical	Y	Bolts (3)	N	Engine Bridal and Hoist
Engine Assist		Y	Electrical	Y	Bolts (3)	N	N
Engine Dress/Heater Hose							
Heater Hose Install	1	Y	Hose	N	Clamps(2)	Clamp Tool	N
Engine Dress/BEC Module							
Secure Electrical Connections	1	Y	Electrical	Y	N	N	N
Secure Electrical Connections and BEC Tray	1	Y	Electrical	Y	Screws (4)	Gun	N

3.1.2   Boeing / GM Human Operations Schema

Human task assessment at Boeing is completed using the Human Operations Schema captured in spread sheet format. It represents all the manual tasks the people perform in manufacturing

operations within plants at Boeing. The first generation of human characterization taxonomy was generated by combining Boeings Human Operations Schema and a General Motors Assembly Ontology (see Appendix B).

3.1.3 MODAPTS

When developing the comprehensive list of assembly task elements, the team started with the existing Boeing, GM and MODAPTS time management systems. A natural extension for iFAB Human Capabilities was to use MODAPTS for performance metrics to account for task element timing. The chart in Table 3 provides the MODAPTS basis used in the Human Capabilities Taxonomy.

Table 3. MODAPTS

MODAPTS		1 MOD = 0.00215 mins	
Group	**Description**	**Code**	**Time (mins)**
Move	Move Small Objects:		
	Finger 1"	M1	0.00215
	Hand 2"	M2	0.0043
	Arm 6"	M3	0.00645
	Whole Arm 12"	M4	0.0086
	Extended Arm 18"	M5	0.01075
	Trunk 30"	M7	0.01505
	Move Large Objects:		
	Hand: 1 Hand	M2	0.0043
	Arm: 1 Hand	M3	0.00645
	Whole Arm: 1 Hand 12" WT=18 LBS; 2 Hands 24" WT=35 LBS	M4	0.0086
	Extended Arm: 1 Hand 18" WT=18 LBS; 2 Hands 35" WT=35 LBS	M5	0.01075
	Trunk: 2 Hands 39" WT=136 LBS	M7	0.01505
Get	Get Small Objects:		
	Contact/Touch	G0	0
	Simple Finger Closing	G1	0.00215
	Complex Finger Closing	G3	0.00645
Put	Put Small Objects		
	To General Location	P0	0
	To Specific Location	P2	0.0043
	To Exact Location	P5	0.01075
	Tear Paper/ Per Tear	P9	0.01935
Load	Load Factor for Small Objects		
	< 4.4 LBS	L0	0
	> 4.4 LBS < 13.3 LBS	L1	0.00215
	> 13.3 LBS < 17.6 LBS	L2	0.0043
Read			
	One Word, General Reading	R2	0.0043

	One Word. Careful Reading/ Reading up to 3 Digits	R3	0.00645
Vocalize			
	For Each Spoken Word	V3	0.00645
Decide/Keystroke			
	For the Unusual Case	D3	0.00645
	Keystroke Using Video Display	D3.3	0.00709
Number			
	Per Item Arranged	N3	0.00645
	Per Item Disarranged	N6	0.0129
Handwrite			
	1 Word Continuous	H21	0.04515
	1 Word Discontinuous	H25	0.05375
	1 Word Upper Case	H35	0.07525
	1 Character Continuous	H4	0.0086
	1 Character Discontinuous	H5	0.1075
	1 Character Upper Case Continuous	H6	0.0129
	1 Character Upper Case Discontinuous	H7	0.01505
Arithmetic			
	Difficult Multiplication	A18	0.0387
	Difficult Division	A24	0.0516
	Addition/Subtraction and Simple Multiplication	A5	0.01075
Walk			
	Walk (x, # of feet)	W_	2.4 * (# feet)
	Walk One Linear Foot	W2.4	0.00516
	Walk One Step	W5	0.01075
	Walk Three Steps	W15	0.03225
	Walk Five Steps	W25	0.05375
	Walk Seven Steps	W35	0.07525
	Walk Eight Steps	W40	0.086
Foot Action			
	Heel Remains on Floor	F3	0.00645
Bend and Arise			
	Hand Goes Below Knee	B17	0.03655
Sit and Stand			
	Sit or Stand	S15	0.03225
	Production Work	S30	0.0645
	Office Work	S48	0.1032
Crank			
	Wrist Up to 3.5" Diam.	C3	0.00645
	Arm Above 3.5" Diam.	C4	0.0086

Juggle			
	Contact/Touch	J0	0
	To Gain Better Control	J2	0.0043
Extra Force			
	A Hesitation	X4	0.0086
Use Tool			
	Finger Motions	U0.5	0.00107
	Hand Motions	U1	0.00215
	Arm Motions	U2	0.0043
	Whole Arm Motions	U3	0.00645
Eye Control			
	Eye Fixation/Eye Travel	E2	0.0043
	Eye Focus	E4	0.0086
Tooling/Process Time			
	Rivets. Small Short Screws	T2	0.0043
	Medium Screws. Deep Tool Powered Prestarts	T4	0.0086
	Long Screws	T8	0.0172

3.1.4   Process Capability Taxonomy

Comprehensive list of human capabilities for assembly tasks was synthesized from GM, Boeing, and MODAPTS IE standards and formalized as HAMML. For a spreadsheet representation of HAMML, see Appendix A.

3.1.5   Human Whole Body and Hand Posture Libraries

The iFAB team was challenged to provide a higher level of abstraction that is more easily used during the early design process. The process was started with hand clearance. Cutkosky's hand grasp model ([23]) was used as a starting point (Figure 2). In addition, 3D hand models (Figure 3) (or "envelopes") for standard grasps were created and placed in library for use (like placing hands on hand tools) in design, manufacturing and assembly feasibility assessments.

 A similar library for whole body postures and associated peak strengths were also created and placed in a library. The 3D whole body models (Figure 4) for standard postures were created and placed in library for use (similar to hand models) in design, manufacturing and assembly feasibility assessments. Using these models requires selection of an appropriate posture for each task assessment. Library postures were obtain from the Siemens tool and they were derived through decades of experience at GM and other Siemens tool users. The 50th % male has been included in the library. Manikin anthropometry is based on ANSUR data [24] from 1988 anthropometric survey of U.S. Army personnel.

3D reach zones (or "work envelopes") and critical dimensions have been pre-calculated for each of the standard postures and the models have been placed in the library for use in design, manufacturing and assembly feasibility assessments. Use of the reach zones requires selection of a standard posture (based on the task and layout).

Approved for public release; distribution unlimited.

In addition to reach zones, **maximum whole body** force when applying load in primary directions (lift up/pull down, push/pull left and right, pull in towards torso, push out away from torso) was pre-calculated for each standard whole body posture. These values were placed in the library.

It's important to highlight two key points:

- Ergonomic limits on acceptable force requirements are generally set lower than the maximum force capability to account for fatigue, and in current state of art, this is based on operational experience.
- To provide design feedback, it is necessary to know how much force is required to perform the operation. This information should come from the META design package.

Figure 2. Hand Grasps

Approved for public release; distribution unlimited.
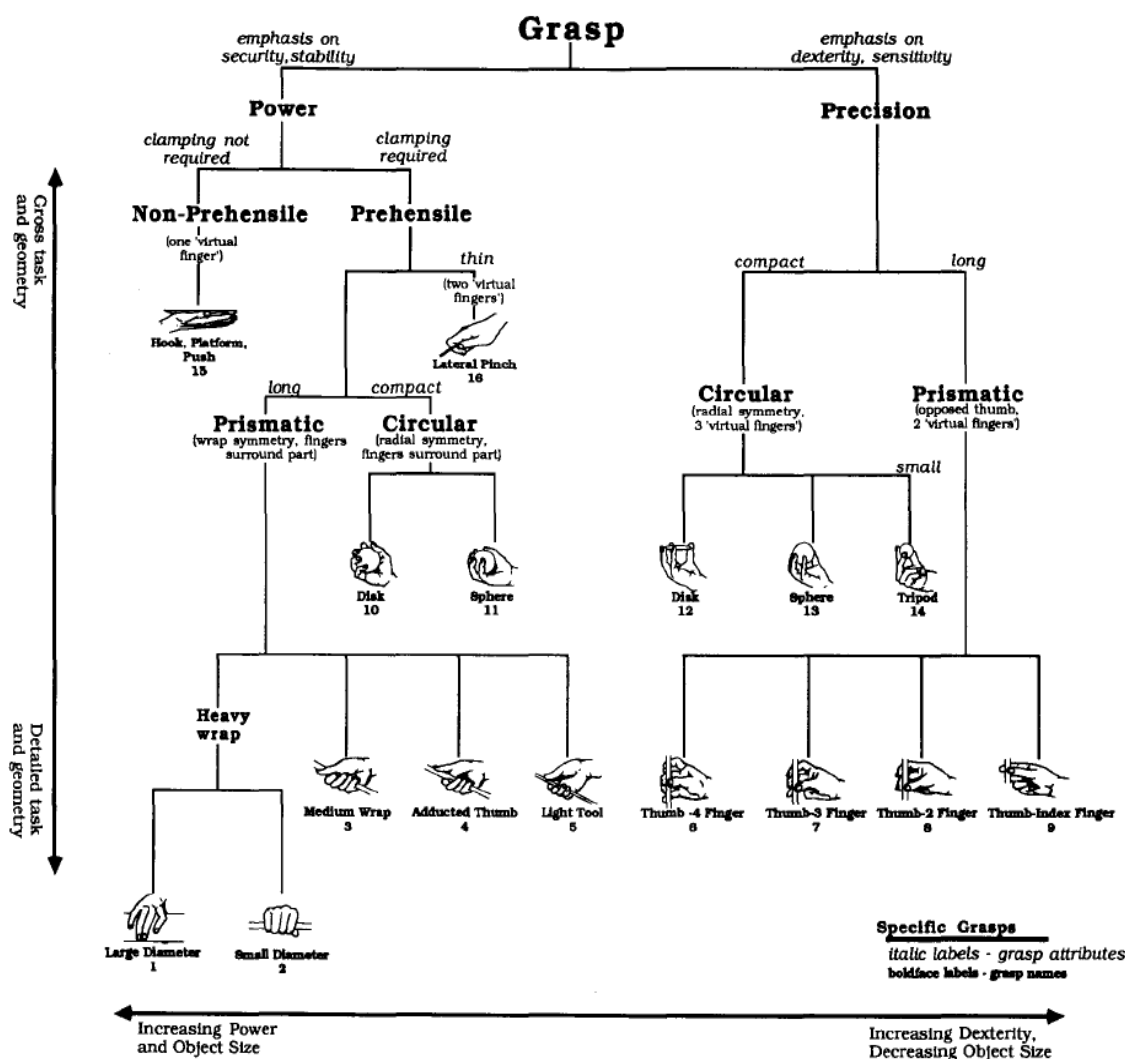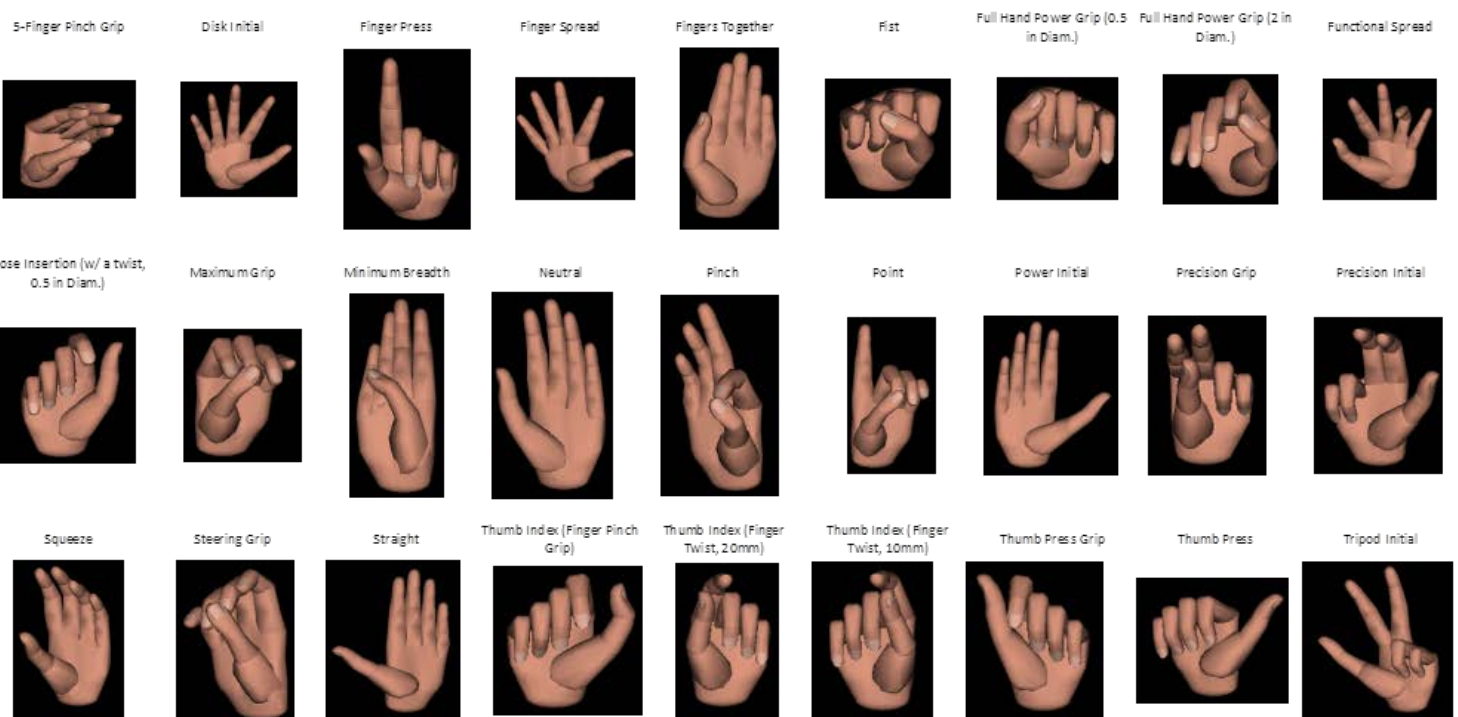
Figure 3. 3D Hand Posture Clearance Zones

5-Finger Pinch Grip | Disk Initial | Finger Press | Finger Spread | Fingers Together | Fist | Full Hand Power Grip (0.5 in Diam.) | Full Hand Power Grip (2 in Diam.) | Functional Spread

Hose Insertion (w/ a twist, 0.5 in Diam.) | Maximum Grip | Minimum Breadth | Neutral | Pinch | Point | Power Initial | Precision Grip | Precision Initial

Squeeze | Steering Grip | Straight | Thumb Index (Finger Pinch Grip) | Thumb Index (Finger Twist, 20mm) | Thumb Index (Finger Twist, 10mm) | Thumb Press Grip | Thumb Press | Tripod Initial

Figure 4. 3D Whole Body Postures

3.1.6 Plant Equipment, Material and Tool Characterization

A listing of plant equipment, materials and tools was generated to capture and associate processes with plant elements (Table 4). Use of plant elements requires the selection of a hand posture to grasp the object while performing the associated task. This task based hand posture was also included in the table for each choice of plant equipment, material and tool.

Table 4. Task based hand posture for plant equipment, material and tools.

Plant Equipment, Material and Tools	Task Based Hand Posture
Adhesive-Glue	Precision Grip
Air Operated Headers Power Tool	Maximum Grip
Air Power (Manual/Automatic Reverse) Driver	Maximum Grip
Air-Hydraulic Headers Power Tool	Maximum Grip
Allen Wrench	Full Hand Power Grip
Battery Powered Drivers	Maximum Grip
Belt Buckle Cover	Precision Grip
Belt Buckle Cover	Precision Grip
Bike	Neutral
Bike	Steering Grip
Brush	Full Hand Power Grip
Bungee (Elastic Strap)	Thumb Index Pinch Grip
CAM-Lock Wrench	Full Hand Power Grip
Cart	Neutral
Cart	Steering Grip
Channel Locks	Maximum Grip
Cleaner	Squeeze
Computer	Neutral
Computer	Precision Initial
Computer	Full Hand Power Grip
Container	Five Finger Pinch Grip
Conveyor	Five Finger Pinch Grip
Crane	Five Finger Pinch Grip
Crane	Neutral
Critical Torque Controlled "Smart" Tools	Maximum Grip
Dispensing gun	Precision Initial
Drift Pin	Maximum Grip
Dunnage	Five Finger Pinch Grip
Ear Plugs /Protection	Precision Grip
Fender Protector Pad (right and left)	Five Finger Pinch Grip

Approved for public release; distribution unlimited.

Fluid Fill Nozzle	Squeeze
Fork truck	Neutral
Fork truck	Steering Grip
Glasses	Precision Grip
Gloves	Precision Grip
Hammer	Full Hand Power Grip
Hard Hat	Precision Grip
Hook Hoist	Thumb Index Finger Pinch Grip
Inline Power Tool	Maximum Grip
Inline Power Tool	Full Hand Power Grip
Label (with Backing)	Precision Grip
Label (without Backing)	Precision Grip
Lift Assist	Thumb Index Finger Pinch Grip
Lift Table	Neutral
Lubricants	Squeeze
Machine	Five Finger Pinch Grip
Mallet	Full Hand Power Grip
Marker	Precision Grip
Mirror	Full Hand Power Grip
Multiple Spindle Air Drivers	Maximum Grip
Multiple Spindle Electrical Transducerized Drivers	Maximum Grip
Oil Pulse Driver	Maximum Grip
One-hander Pliers	Full Hand Power Grip
Pallet	Five Finger Pinch Grip
Paper	Five Finger Pinch Grip
Paper	Precision Grip
Pistol Grip Power Tool	Precision Initial
Pistol Grip Tool	Full Hand Power Grip
Plews Oiler	Squeeze
Pliers	Squeeze
Prop Rod	Five Finger Pinch Grip
Protective Suit	Precision Grip
Prybar	Full Hand Power Grip
Punch	Full Hand Power Grip
Putty Knife	Full Hand Power Grip
Rag	Fist

Ratchet (box end) Wrench	Full Hand Power Grip
Right Angle Power Tool	Full Hand Power Grip
Right Angle Power Tool	Maximum Grip
Ring Cover	Precision Grip
Rivet Gun	Maximum Grip
Roller Tool	Full Hand Power Grip
Sand Paper	Thumb Press
Scissors	Precision Grip
Screwdriver	Full Hand Power Grip
Seal Tool	Full Hand Power Grip
Shipping Caps	Pinch
Side-cutter	Squeeze
Single Spindle Air Drivers (Manual/Auto reverse)	Maximum Grip
Single Spindle Air Transducerized Drivers	Maximum Grip
Single Spindle Electrical Transducerized Drivers	Maximum Grip
Sleeves	Precision Grip
Snap Gage	Five Finger Pinch Grip
Socket Handle	Full Hand Power Grip
Socket Wrench (Tee-handle)	Full Hand Power Grip
Solder Gun	Precision Initial
Speed (open end) Wrench	Full Hand Power Grip
Spray Bottle (Pump)	Squeeze
Spray Gun	Precision Initial
Squeegee	Five Finger Pinch Grip
Staple Gun	Squeeze
Tape	Precision Grip
Tape (with Backing)	Precision Grip
Tape (without Backing)	Precision Grip
Ties (zip or ban)	Precision Grip
Tote/ Bin for Kitting	Five Finger Pinch Grip
Tube Nut Runners	Full Hand Power Grip
Tube Nut Runners	Maximum Grip
Tugger	Neutral
Tugger	Steering Grip
Watch Cover	Precision Grip
Water	Squeeze
Work Boots	Precision Grip
Yankee (Push) Screwdriver	Full Hand Power Grip

3.2   Characterization of the Manufacturing Machine and Tool Space

The following section provides the research captured to characterize the machine and tool space of the manufacturing domain. Within the context of manufacturing, numerous other facets of machining processes impact the fabrication, so there are also included characterization of tolerance and machining error.

This section discusses the background information that provides the basis for a manufacturing machine schema. The MCPML focuses on the manufacturing machine processes in Table 5.

Table 5. Manufacturing Machine Processes in MCPML

Process	Description
Mass-change	Remove or add material, such as chip making: shaping, turning, milling, drilling, sawing, punching, broaching, grinding, welding, fastening, laser cladding or a plating operation
Phase-change	Produce a solid part from liquid material, such as metal casting, infiltration of composites, and injection molding of polymers
Structure-change	To alter material mechanical properties, such as mechanical treatment and thermal treatment, such as laser heat treatment, costing, laser cladding (also mass change)
Deformation-change	Transform solid materials from one shape into another, such as forging, bending, rolling, drawing, extrusion processes
Consolidation	Assembly of smaller objects into a single product in order to achieve a desired geometry, structure, or property, such as powder processing, powder injection molding, some steps in composites manufacturing
Integrated	Combine more than one specific unit process into a single piece of equipment or into a group of work stations that are operated under unified control
Assembly	Manufacturing process in which parts are added to a product in a sequential manner using optimally planned logistics to create a finished product much faster than with handcrafting-type methods
Metrology and Quality Control	Processes used to measure the accuracy of the parts and components needed in the manufacturing process, and contribute in improving productivity and efficiency of production processes

Each process can be generally characterized into input material or part, with the proper product and manufacturing process data, and other resources such as workforce, equipment, and energy, and resulting in process output. This includes product and the process effect on environment.
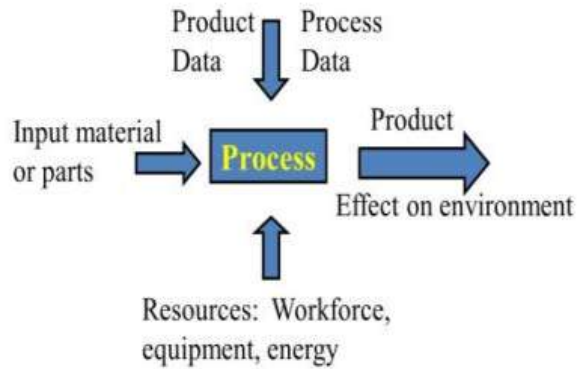
Figure 5. General Process Information and Materials Flow for each Typical Process
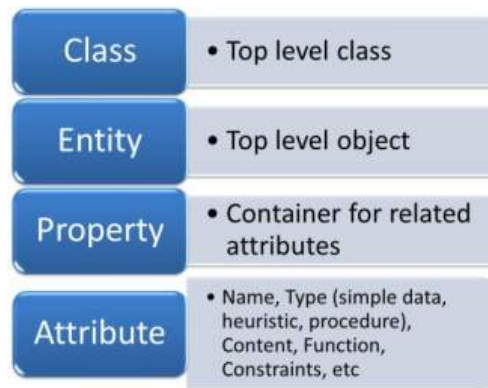
Figure 6. The "Object Hierarchy" Used to Organize Objects, Properties, Entities, Constraints

3.2.1 Hierarchy of Machine and Tool Description

Although tree-based structures impart decision making ability, their volume of data sometimes makes the structure redundant. To overcome these issues we build a dynamic structure that is more flexible than a tree-based structure. The new structure is based on basic building blocks like entity, property and attribute. These blocks are designed into modules which contain all the requisite information about a particular topic (e.g. milling, tool data will be modules). These modules are connected to the structure based on the concept of inheritance as in the object-oriented concept. We define a representation hierarchy that can be adapted for general manufacturing processes in the library, such as mass change, phase changes, structure changes, etc. The "object hierarchy" is used to organize the objects, properties, entities, functions, constraints, etc., as shown in Figure 6.

The MCPML library schema is designed to support the extraction of the individual physical attributes associated with a product and their incorporation in manufacturing modules which will help with decision-making required to manufacture the part. The library is a repository of large amounts of data about the manufacturing environment of a foundry. It is being designed as an aid to not just the process planner, but also the designer and the machinist.

Approved for public release; distribution unlimited.

We develop an interactive software environment that will not just display relevant information but it also acts as a logical system that shortlists tools and machines that a planner can use to draw up a process plan. For the design engineer it provides the ability to identify manufacturing and assembly constraints of a part and alter the design accordingly saving significant time and money. It is used by the machinist to efficiently identify process parameters to obtain the best possible surface finish and tolerance using the resources available in the foundry environment.

A service-based front end user interface is added onto the library to suit the feature extraction process and also for the evaluation of the tolerance requirements. This interface takes input from the customer requirements and gives the output utilizing the attributes from the manufacturing library. Figure 7 depicts the various modules in the manufacturing library. The data flow schema among these modules will be discussed in detail in the forthcoming sections.
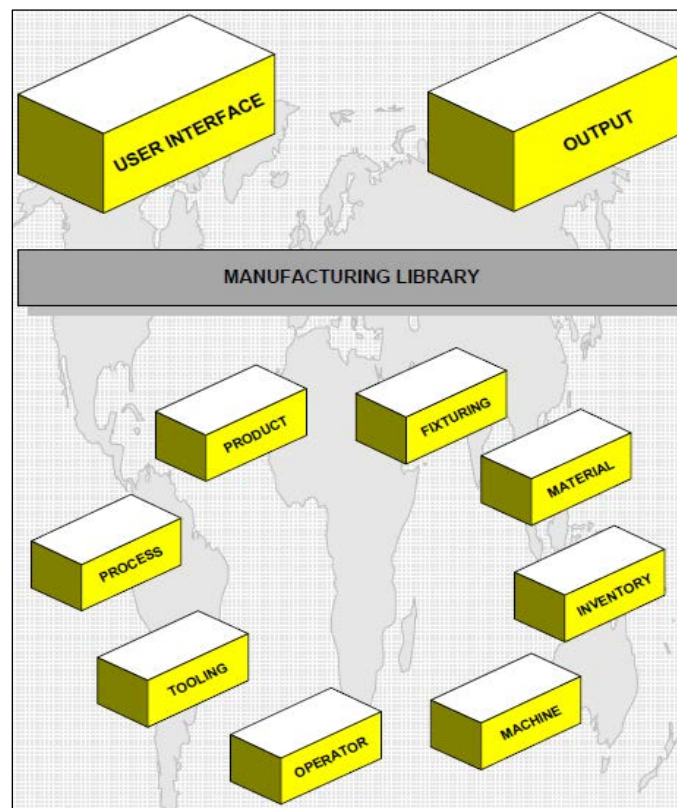
Figure 7. Manufacturing Library Indicating Hierarchy over Modules

3.2.2 Development of individual modules

The entire manufacturing module is divided into individual factories which in turn are divided into manufacturing work cells consisting of machines as shown in Figure 8. The entire manufacturing environment is split up into individual modules with the corresponding data embedded inside. Schemas are developed for the individual modules and are detailed in the following sections.
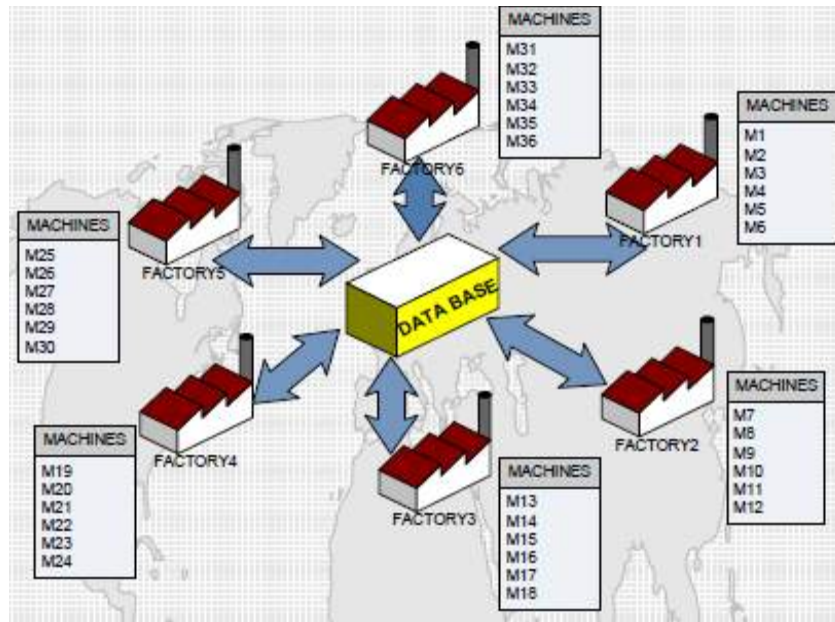
Figure 8. Factory Layout Aggregation of Manufacturing Modules

3.2.2.1 Machine module

The schema of the machine module was developed taking into consideration the detailed specifications obtained from technical literature of various machines available at MS&T and online literature. More than twenty machine specifications were studied so as to ensure the inclusion of all operating parameters and the specifications of each individual machine.

All the tools in the machine library are included in the machine module and the detailed tool details are included in the tooling and material module which will be discussed later in the report. The optimum operating conditions are obtained from the information provided by the manufacturer. Economic and management information are also provided so as enable the calculation of the cost and time of the operations required.

Table 6 shows the machine module developed for this project in detail. Its includes details on important attributes that not only help identify and characterize the machine but also help in cost and time computations that will discussed in detail in forth coming sections.

Table 6. Machine Module

1	Machine	1.4.2	Accuracy (Inches)
1.1	Process Sets	1.4.3	Resolution (Inches)
1.1.1	Process 1	1.4.4	Load (lb)
1.1.2	Process 2	1.4.5.1	Min
1.2	Facilities	1.4.5.2	Max
1.2.1	Physical Properties	1.4.5.3	Horsepower
1.2.1.1	Machine Size (inches)	1.4.6	Power Consumption

Approved for public release; distribution unlimited.

1.2.1.1.1	Along X-Axis		1.5	Set of Tools
1.2.1.1.2	Along Y-Axis		1.5.1	Number of Tools
1.2.1.1.3	Along Z-Axis		1.6	End Effector sets
1.2.1.2	Table Size (inches)		1.6.1	Module Name
1.2.1.3	Workpiece size		1.6.2	Tool Change Time
1.2.1.5	Max. Workpiece Weight		1.7	Management Info
1.2.1.6	Spindle to Table Distance(In.)		1.7.1	Maintenance Info
1.2.1.7	Spindle Diameter(Inches)		1.7.2	Name (Serial Number)
1.2.1.8	Table Load Capacity		1.7.3	Vendor Info
1.2.2	Utilities Requirement		1.7.4	Operator Info
1.2.2.1	Current rating (Amp)		1.7.5	Manufacture Date
1.2.2.2	Voltage rating (V)		1.7.6	Consumable
1.2.2.3	Compressed air req. (psi)		1.7.6.1	Lubricant
1.2.3	Noise Level (db)		1.7.6.2	Coolant
1.3	Axes Format		1.7.7	Availability
1.3.1	Axis 1		1.7.7.1	Availability statues(Y/N)
1.3.1.1	Axes Name		1.7.7.2	Availability Date
1.3.1.2	Dependence		1.7.7.3	Availability Time
1.3.1.3	Motion type		1.7.8	Machine ID
1.3.1.4	Motion axis		1.8	Economic Info
1.3.1.5	Travel		1.8.1	Billing rate (USD / hr)
1.3.1.6	Travel speed (IPM)/		1.8.2	Purchase price (USD)
1.4	Motion		1.8.3	Life time maintenance costs (USD)
1.4.1	Repeatability (Inches)		1.8.4	length of the loan (years)
1.4.1.1	X - Axis		1.8.5	Annual maintenance costs
1.4.1.2	Y - Axis		1.8.6	Price of electricity / KW
1.4.1.3	Z - Axis		1.8.7	Projected machine hours / year
1.4.1.4	A- Axis		1.8.8	cost of consumables (USD/year)
1.4.1.5	B- Axis		1.8.9	Depreciation Period (% per year)

3.2.2.2 Operator Module

The operator module is developed on similar lines to the machine module schema. The various skill sets, physical capabilities and restrictions along with the required general information are included into the comprehensive schema.

- Skill sets: These include the various operations that an operator can perform. In other words it can be stated as various job classification codes associated with the particular operator.
- Physical: These consist of various physical capabilities of the operator. This data is based on the anthropometric characteristics (gender, size percentile, and dextrosity).

- General information: In addition to all these details, basic information such as the name, designation and contact details etc. are also included in the operator module.

Table 7. Operator Module

1	Operator		1.1.9	Maintenance
1.1	Process		1.1.9.1	Skill level 'S' (1-5)
1.1.1	General		1.1.9.2	Experience
1.1.1.1	Skill level 'S' (1-5)		1.1.9.2.1	Equipment
1.1.1.2	Experience		1.1.9.2.2	Hours
1.1.1.1.1	Equipment		1.2	Anthropometric percentile
1.1.1.1.2	Hours		1.2.1	S/M/L (5%/50%/95%)
1.1.2	Machinist		1.3	Gender
1.1.2.1	Skill level 'S' (1-5)		1.3.1	M/F
1.1.2.2	Experience		1.4	Physical Restriction
1.1.2.2.1	Equipment		1.4.1	N LADDER (Y/N)
1.1.2.2.2	Hours		1.4.2	N LIFT (Y/N)
1.1.3	Millwright		1.4.3	N WALK (Y/N)
1.1.3.1	Skill level 'S' (1-5)		1.4.4	N CLIMB (Y/N)
1.1.3.2	Experience		1.5	Management info
1.1.3.2.1	Equipment		1.5.1	Basic info
1.1.3.2.2	Hours		1.5.1.1	Type: (Regular / Visitor)
1.1.4	Electrician		1.5.1.2	ID Badge info
1.1.4.1	Skill level 'S' (1-5)		1.5.1.2	Employee ID
1.1.4.2	Experience		1.5.1.3	Expiration date
1.1.4.2.1	Equipment		1.5.1.3	US Citizen - (Y/N)
1.1.4.2.2	Hours		1.5.2	Wage (USD/hr)

3.2.2.3 Tooling Module

While designing the tooling module we assumed that it is easier to buy machine tools from manufacturers than to make them in house. When tools are purchased from a manufacturer we can obtain certain information from the manufacturer about the tool and optimum cutting conditions.

We have analyzed considerable literature and shortlisted certain information that every manufacturer provides with their tools as shown in Table 8.These will be the attributes of the module and they will provide a reference for operators.

Every tool is used specifically for a particular operation and will be related to a particular process in the process module. This is defines a relation between the tooling module and the process module.

Table 8. Tooling Module

1	Tool		1.4.1.3	No. of flutes
1.1	Set of Processes		1.4.1.4	Helix angle (degrees)
1.1.1	Process 1		1.4.1.5	Corner radius
1.1.2	Process 2		1.4.1.6	Reach length
1.2	Recommended cutting material		1.4.1.7	Diametric tolerance
1.2.1	Material 1		1.4.2	Material
1.2.2	Material 2		1.4.2.1	Tool material
1.3	Operating conditions		1.4.2.2	Coat material
1.3.1	Material 1		1.4.2.3	Tool hardness (Vickers)
1.3.1.1	Optimal feed (ipr)		1.5	Optimum tool life
1.3.1.2	Optimal speed (SFM)		1.6	TOLERANCE MODEL
1.3.1.3	Chipload per flute (inches)		1.7	Management info
1.3.1.4	Coolant		1.7.1	Ordering number
1.3.2	Material 2		1.7.2	Vendor info
1.3.1.1	Optimal feed (ipr)		1.7.3	Serial number
1.3.1.2	Optimal speed (SFM)		1.7.4	Manufacturing date
1.4	Tool details		1.8	Economic info
1.4.1	Tool geometry		1.8.1	Price of tool (USD)
1.4.1.1	Overall Length (Inches)		1.8.2	Tool usage (hrs)
1.4.1.2	Shank diameter (Inches)		1.8.3	Maintenance costs (USD)

3.2.2.4 Tool Holder Module

The tool holder module contains the data of all the tooling that a holder can accommodate and the machine on which it can be used. It also contains the geometric specifications of the tool holder and also the clamping force requirements which in turn will be used for specifying any limitations on the part of operator. Table 9 shows the tool holder module in detail.

Table 9. Tool Holder Module

1	Tool holder
1.1	Tools list
1.1.1	Tool 1
1.2	Facilities
1.2.1	Physical properties
1.2.1.1	Weight

1.2.1.2	Material	
1.2.1.3	Material hardness (HRC)	
1.2.1.4	Size	
1.2.1.2.1	Taper standard	
1.2.1.2.2	Collet length	
1.2.1.2.3	Collet diameter	
1.2.1.2.4	Projected length	
1.3	Operating parameters	
1.3.1	Collet locking range	
1.3.1.1	Min	
1.3.1.2	Max	
1.3.2	Clamping type at insert	
1.3.3	Clamping force	
1.3.4	Theoretical drive spring life (Cycles)	
1.3.5	Recommended lubrication interval (min)	
1.3.6	Cutting fluids capacity	
1.6	Economic info	
1.5.1	Purchase price (USD)	
1.5.2	Life time maintenance costs (USD)	
1.5.3	Annual maintenance costs (USD/year)	
1.5.4	Projected machine hours / year	

3.2.2.5 Workpiece Module

This module (Table 10) contains historical data on the fabrication of workpieces gathered over time in the foundry (factory) environment as a way of managing library information pertaining to the more workpiece specific information about tooling dynamic performance. This includes data pertaining to the tool life history, tool failures and the causes for the same.

We initially tried to populate the library tooling module with data such as comparative hardness between tool and work piece and other factors important to cutting that could be used to predict fabrication performance such as hot hardness, impact strength, wear resistance, chemical inertness etc. Collection of tools data based on the hardness proved to be unsuccessful due to difficulty in obtaining information for factors such as wear resistance, hardness and inertness etc. Accordingly, we adopted a history-based approach to predicting tool performance to account for all these shortcomings. This experience based approach demands the operator to fill in the data periodically. This type of schema proves to be successful wherein the experience gained is used in the decision making process.

In this approach, each piece of tooling should have the manufacturer's data fed into the tooling schema. As the tooling is used to fabricate workpieces, the characteristics of the tooling in interaction with various work materials under various operating conditions are populated overtime. For one particular tool the following parameters are included in the library:

- Various work piece materials
- Features machined in the past
- Feature dimensions
- Optimum tool life
- Cutting parameters
- Tolerance criteria
- Record of failures and their respective causes
-

The operator is asked to fill out a form whose information will then be provided to other operators in future to aid in their decision making process.

Table 10. Workpiece Module

1	Tool
1.1	Instance 'n'
1.1.1	Machine
1.1.2	Work piece
1.1.2.1	Material
1.1.2.2	Hardness (Rockwell)
1.1.3	Feature details
1.1.3.1	Feature machined
1.1.3.2	Dimensions (Inches)
1.1.3.2.1	X
1.1.3.2.1	Y
1.1.3.2.3	Z
1.1.4	Tool usage (hrs)
1.1.5	Cutting conditions
1.1.5.1	Speed (RPM)
1.1.5.2	Feed rate (IPM)
1.1.6	Failure report
1.1.6.1	Failure (if any) - Y/N
1.1.6.2	Cause of failure
1.1.7	Tolerance
1.1.8	Management info
1.1.8.1	Operator
1.1.8.2	Project Number
1.1.8.3	Date

3.2.2.6 Hand Tool Module

This schema has been developed taking into consideration the entire Atlas Copco hand tools library. The hand tools library has been thoroughly investigated in order to obtain the performance characteristics and geometrical limitations of the hand tools. The following are various hand tools in the library taken into account while developing the library:

- Pneumatic assembly tools
- Electric assembly tools
- Quality assurance in tightening
- Grinders
- Percussive tools
- Drills
- Automatic drilling and tapping units
- Airline accessories
- Air motors
- Hoists and trolleys

The operating parameters, motion and the accessibility sections of the schema facilitate decision making in terms of the usage and the suitability of the particular hand tool in a job. The physical attributes include the envelope size, weight and the utility requirements in the operation of the hand tool. Ordering numbers and CAD files are also included to aid the purchase department in the acquisition or service of the tools. Table 11 shows the hand tool module in detail.

Table 11. Hand Tool Module

1	Hand tool		1.3.6	Max. cutting depth (Inches)
1.1	Process		1.4	Motion
1.1.1	Process 1		1.4.1	Torque range (in lb)
1.2	Facilities		1.4.2	Max Torque (in lb)
1.2.2	Physical properties		1.4.3	Speed
1.2.2.1	Weight (Lb)		1.4.3.1	At max. output
1.2.2.2	Size		1.4.3.2	At free speed (RPM)
1.2.2.2.1	Chuck/Collet		1.4.5	Gear ratio
1.2.2.2.2	Air hose inlet (Inches)		1.4.6	Length feed
1.2.3	Utilities requirements		1.4.7	Stroke
1.2.3.1	Electrical		1.5	Accessibility
1.2.3.1.1	A/C power rating		1.5.1	CS distance (Inches)
1.2.3.1.1.1	Voltage (V)		1.5.2	Tool length envelope (Inches)
			1.6	Set of tools
1.2.3.1.1.2	Current (Amp)		1.6.1	Chuck/Collet
			1.6.1.1	T1
1.2.3.1.2	Battery		1.6.1.2	T2

1.2.3.1.2.1	Type		1.6.2	Hook up kit
1.2.3.1.2.2	Voltage		1.6.2.4	Weight (lbs)
			1.7	End effector sets
1.2.3.1.2.3	Capacity		1.7.1	DRIVE (Bit)
1.2.3.2	Pneumatic		1.7.1.1	Square
1.2.3.2.1	Air consumption rate (l/s)		1.7.1.2	Hexa
1.2.3.2.1.1	At max output		1.7.1.3	Ratchet
			1.8	Management info
1.2.3.2.1.2	At free speed		1.8.1	Model number
1.2.3.2.2	Rec. hose size (Inches)		1.8.2	Ordering number
1.3	Operating parameters		1.8.2.1	Hand tool
1.3.1	Operating temperature (Deg. C)		1.8.4	Manufacturing date
1.3.2	Sound levels (db)		1.8.5	CAD Drawing file
1.3.3	Max. power output (KW)		1.9	Economic info
1.3.4	Max. operating pressure (Kpa)		1.9.1	Billing rate (usd/hr)
1.3.5	Measured vibration value (m/sec2)		1.9.3	Annual maintenance costs
			1.9.6	cost of consumables (USD/year)
			1.9.7	Price (usd)

3.2.2.7 Inventory module

The inventory module (Table 12) was created to check the availability of the raw material for processing in the foundry. This also addresses the shape and quantity of the stock. The stock may be located at different locations of the foundry facility to go through the planned production scheduling. This module helps in tracking of the costs related to ordering, logistics.

Table 12. Inventory Module

1	MATERIAL (1 - n)
1.1	Lot (1- n)
1.1.1	Shape
1.1.1.1	Cylindrical (Y/N)
1.1.1.2	Cuboidal (Y/N)
1.1.1.3	Other polygonal (Y/N)
1.1.2	Type
1.1.2.1	Hollow (Y/N)
1.1.2.2	Solid (Y/N)
1.1.2.3	Porous (Y/N)
1.1.3	Objective data
1.1.3.1	Quantity
1.1.3.2	Weight
1.1.3.3	Volume
1.1.3.3.1	X - size
1.1.3.3.2	Y - size
1.1.3.3.3	Z - size
1.1.3.4	Surface area (occupied)
1.1.4	Supply (1 - n)
1.1.4.1	Ordering number
1.1.4.3	Supplier info
1.1.4.3.1	Address
1.1.4.3.2	Contact info
1.1.4.3.2.1	Phone number
1.1.4.3.2.3	E - Mail
1.1.5	CAD models / drawings
1.1.5.1	Views
1.1.5.2	3D CAD File
1.1.6	Management data
1.1.6.1	Name
1.1.6.1.1	Material
1.1.6.3	Item cost per unit

3.2.2.8 Fixture Module

Table 13 shows the fixture module developed for the iFAB project. Operator assistance on fixtures was provided through the historical data accumulated in the library. The historical data takes into account all possible attributes related to the fixtures that have already been used in similar instances. An instance 'n' is considered, where the details of the fixture that has been used previously is provided.

A method of fixture identification with various views such as isometric view, front view, top view and one side view is provided. An option to provide directly a 3D CAD file is also included. The properties of the fixture such as mounting type, weight of the fixture, material, clamping faces, and position with respect to the machine table are provided.

The position of the fixture is defined with respect to the center of the machine table. A co-ordinate axis is assumed at the center of the machine table and the distance of the fixture in each direction is taken with respect to this co-ordinate system. To avoid redundancy only properties of the part which are necessary for the fixture module and haven't been covered in the part module are considered. The part location is also specified with respect to the same co-ordinate system.

After the part is placed on the fixture, the faces of the part that are available for machining are noted down. The focus is to provide information about re-fixturing and repositioning. The operating parameters of the fixture cover the clamping force required, maximum loaded weight. The setup time, manufacturing time and the usage time (total lifetime operation of fixture in hours) is provided. Finally, the finished product information and the management and economic information are also included. This all inclusive module provides data supporting locating, clamping, and re-positioning of a product.

The fixtures covered by this module are traditional bespoke fixtures. Since bespoke fixtures are by definition created on demand at need, the fixtures in the library represent those existing fixtures available for reuse. The same module can also be used to represent fixtures created for a specific design which would then be available for later reuse. This module does not directly cover the elements of a modular fixturing system.

Table 13. Fixture Module

1	Fixture
1.1	Instance 'n'
1.1.1	Objective data
1.1.1.1	Name and Fixture ID
1.1.1.2	Project Name and info
1.1.1.3	Project Duration
1.1.2	CAD models / drawings
1.1.2.1	Views
1.1.2.1.1	Isometric / Trimetric
1.1.2.1.2	Front / Back
1.1.2.1.3	Top / Bottom
1.1.2.1.4	Left / Right
1.1.2.2	3D CAD File
1.1.2.2.1	Part Files
1.1.2.2.2	STL and mesh file
1.1.2.2.3	Exchange formats
1.1.2.3	Attachments
1.1.2.3.1	Supports
1.1.2.3.2	Locators
1.1.2.3.3	Others
1.1.3	Physical Properties

1.1.3.1	Fixture Properties
1.1.3.1.1	Weight
1.1.3.1.3	Material
1.1.3.1.4	Position wrt machine table
1.1.3.1.4.1	X
1.1.3.1.4.2	Y
1.1.3.1.4.3	Z
1.1.3.2	Contact Faces
1.1.3.2.1	With machine table
1.1.3.2.1.(1-n)	Face (1 - n)
1.1.3.2.2	With part
1.1.3.2.2.(1-n)	Face (1 - n)
1.1.3.3	Part
1.1.3.3.1	Part Location
1.1.3.3.1.1	X
1.1.3.3.1.2	Y

3.2.3   Tolerance

Tolerance is a tool used by product designers to control the geometric and dimensional variation of a product. It is the amount of variation permitted on the dimensions of machine parts. Two types of tolerances, conventional and GD&T, may be used in drawings. The conventional tolerance specification may use (i) limit dimensioning and (ii) plus/minus dimensioning, Figure 9. In the current GD&T tolerancing standards, workpiece precision is expressed through (i) size tolerances and (ii) geometric tolerances (including form, orientation, and location tolerances). Compared with conventional tolerance specifications, GD&T is superior for specifying the geometric variations of a product. It requires a geometric control symbol, tolerance value, and datum(s). Figure 10 (a) shows an example of the GD&T feature control frame, which indicates the type of geometric control for the machining feature, the tolerance for the control, and the related datum plane. Figure 10 (b) shows an example of positioning tolerance using two datums (A and B). The datums determine the orientation and location of the tolerance zone characterized by the tolerance value t in which the axis of the hole can move; and limit dimensioning strategy is used to determine the diameter of the hole. A complete list of ANSI/ASME geometric control symbols can be found in [9] (see also Appendix D Figure D-1).

(a)   (b)
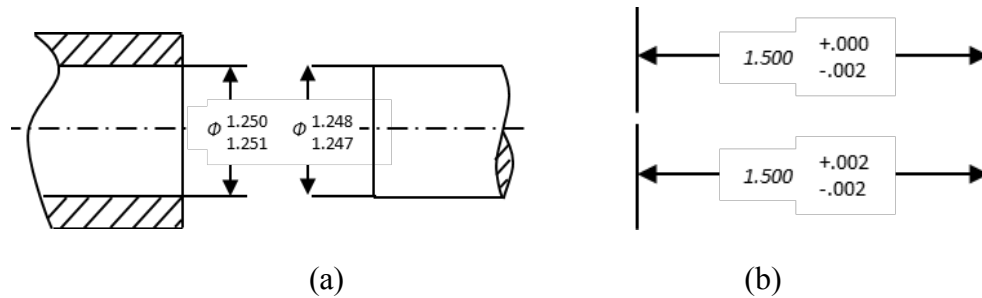
Figure 9. Conventional tolerance on dimensioning (a) limit dimensioning and (b) plus/minus dimensioning.

(a)   (b)
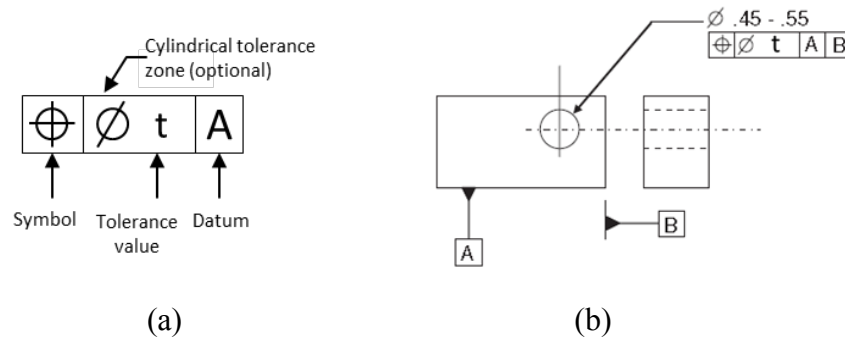
Figure 10. Geometric dimensioning and tolerancing (a) feature control frame for GD&T and (b) dimension and positioning tolerance for a hole.

3.2.3.1 Two levels of tolerance evaluation

Process planning has two distinguished levels, process-level planning and operation-level planning. At the process-level, planning is concerned with identifying the main tasks and their best sequence and the type of manufacturing processes. Operation-level planning details process parameters, required tools and setups, process time and resources. Accordingly, tolerance analysis consists of process-level tolerance evaluation and operation-level tolerance evaluation. In process-level tolerance evaluation, the typical ranges of accuracy are mapped to certain manufacturing processes, which can be found in a machinery book. The operation-level tolerance evaluation is based on machining features and intends to answer the question whether selected machine tool, cutting tool, fixture devices will satisfy the design tolerance.

3.2.3.1.1   Process-level tolerance evaluation for machining

A general guide to machining processes that may normally be expected to produce work within the tolerances indicated by the International Tolerance (IT) grades given in ANSI b4.2-1978 is shown in Table 14. This tolerance grade specifies tolerances with associated manufacturing processes for given dimensions. The tolerance associated with a machining process is calculated as follows (ISO 286)

$$T = 10^{0.2(IT-1)}(0.45\sqrt[3]{D} + 0.001D)$$

(1)

where T = Tolerance in micrometers, D = Dimension in millimeters and IT = International Tolerance grade (a positive integer) for a machining process.

For instance, according to Table 14, the milling process has IT grades ranging from 10 to 11. If the process planner wants to check the tolerance of a dimension that is 500 millimeters, they can calculate it by letting IT = 10, 11 and D = 500. If the design tolerance is smaller than the tolerance calculated with IT = 10, it means the process is unable to produce the required tolerance. If the design tolerance is larger than the tolerance calculated with IT = 11, it means that the process can definitely satisfy the required tolerance. If the design tolerance is between the tolerances calculated when IT is 10 and is 11, an operation-level analysis will be needed to determine whether the process can satisfy the tolerance requirement.

Table 14. Relation of machining processes to IT Tolerance Grades (Oberg et al. 2008).

	IT Grades							
	4	5	6	7	8	9	10	11
Lapping & Honing	■	■						
Cylindrical Grinding		■	■					
Surface Grinding		■	■	■	■			
Diamond Turning		■	■	■				
Diamond Boring		■	■	■				
Broaching		■	■	■				
Powder Metal sizes			■	■				
Reaming			■	■	■			
Turning				■	■	■	■	■
Powder Metal sintered				■	■	■		
Boring				■	■	■		
Milling							■	■
Planing & Shaping							■	■
Drilling							■	■
Punching							■	■
Die Casting								■

3.2.3.1.2 Operational-level tolerance evaluation for machining

Figure 11 shows a block diagram for operation-level tolerance evaluation for machining processes, the inputs of which include machining features, tolerance specifications, and process parameters. Based on the machining features and tolerance specifications, manufacturing resources including machine tool, cutting tool, workpiece, and fixture can be selected from the manufacturing process library. Process parameters are also specified, which include feed rate, spindle speed, and depth of cut. According to the toleranced machining features and the tolerance types, tolerance zone constraints are set up. The constraints are evaluated using the agglomerated error factors from different error sources. If the constraints are satisfied, the machining process is considered to be capable of satisfying the tolerance requirement.

Tolerance evaluation is based on actual machining error synthesis after analyzing the tolerance zone of the each machining feature. Section 3.2.3.2 discusses the mapping relationship between different types of machining features and tolerances. The displacement parameters for machining features are also defined. A systematic way of evaluating various types of tolerances

specification for different machining features is presented in Section 3.2.3.3. Section 3.2.3.4 describes the estimation from different error sources, and combining the various errors in the nominal feature coordinate system. Section 3.2.3.5 presents an illustrative example of the methodology.
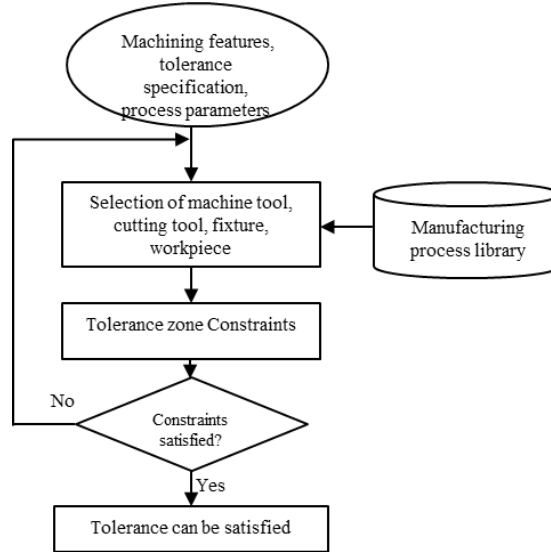
Figure 11. Diagram of process tolerance capability evaluation.

3.2.3.2 Tolerance evaluation based on machining features

For machining operations, features are the individual geometric shapes that are cut into the stock to form the part. Since the surface generated by machining operations can be extremely complex, it is easier just consider typical machining features. Typical features encountered in manufacturing consist of hole, pocket, slot, step, fillet, and chamfer [11].

3.2.3.2.1   Machining feature and tolerances

For a machining feature, there is only finite number of ways to specify tolerances. For example, for a hole, the most popular ways to specify tolerances include dimensions, axis perpendicularity, axis parallelism, and axis position. Figure 12 shows different tolerance specifications mapped with various machining features. In the following sections, tolerance evaluation schemes for different combination of machining features and tolerances will be defined systematically.
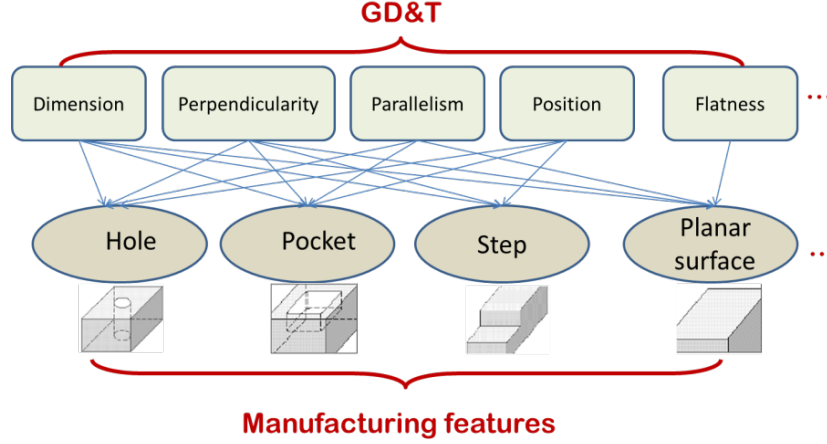
Figure 12. Manufacturing features and GD&T.

3.2.3.2.2 Displacement parameters

For a machining feature, the amount of displacement with respect to a reference frame will depend on six independent parameters; three translation parameters and three rotation parameters. And the displacement for a machining feature can be represented as:

$$T = \{ \delta_x, \delta_y, \delta_z, \varepsilon_x, \varepsilon_y, \varepsilon_z \} \tag{2}$$

where $\delta_x, \delta_y, \delta_z$ are positional displacements in x, y and z axis, respectively, and $\varepsilon_x, \varepsilon_y, \varepsilon_z$ are rotational displacements in x, y and z axis, respectively.

The displacements of the machining feature with respect to the nominal feature coordinate system are caused by various error sources. In Section 3.2.3.3, the constraints are set up for the displacements of the machining features with different tolerance specifications. And in Section 3.2.3.4, the displacements of the machining feature resulting from different error sources are analyzed.

3.2.3.3 Tolerance and constraints

The tolerance of a feature defines the zone within which every point of the feature is to be contained. According to the toleranced feature and the manner it is dimensioned, the tolerance zone can be the space between two parallel planes, the space within a cylinder, and the space within a parallelepiped [1].

For a machining feature, in order to let it stay within the tolerance zone, a set of constraints on the displacements need to be established. According to [3], constraints are established for different tolerances including orientation tolerances, position tolerances, and dimension tolerances for typical machining features.

3.2.3.3.1 Orientation tolerances

Tolerance zone for the two types of orientation tolerance (parallelism, perpendicularity) can be defined by two planes parallel to each other. Figure 13 shows two examples of perpendicularity

Approved for public release; distribution unlimited.

and parallelism whose tolerance zones are defined by two parallel planes as illustrated by Figure 13 (c).

(a)   (b)

(c)   (d)
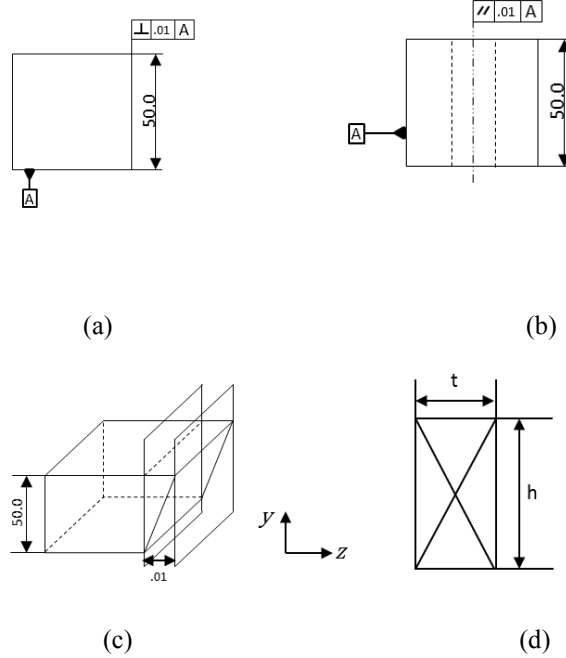
Figure 13. Orientation tolerance (a) Perpendicularity (b) parallelism (c) an extreme case of a toleranced feature and (d) tolerance zone.

In the case of Figure 13 (a), the translation along normal direction of the toleranced plane should be constrained by dimension tolerance, and the rotation along y-axis should be constrained by another geometrical tolerance. In other words, only orientation in x direction needs to be constrained. In this case, the error sources causing the orientation displacement in x direction are taken into consideration. Figure 13 (c) shows the tolerance zone, and one worst case for the planar surface. The two worst cases are illustrated by Figure 13 (d), and the range of the orientation deviation in x direction is

$$-\frac{t}{h} \le \varepsilon_x \le \frac{t}{h}$$ (3)

where t = 0.01 and h = 50.

Figure 14 (a) shows an example about the tolerance zone for the axes of two cylinders in parallel with each other. The parallelism of the axis of the cylinder with respect to the datum axis remains invariant under rotation and translation about its own axis. Also, translation along x and y-axis will not affect the parallelism. So only rotation about x and y direction needs to be constrained.

(a)   (b)
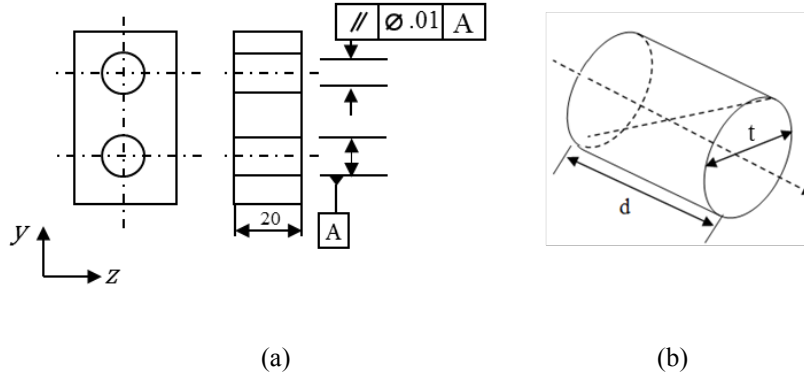
Figure 14. Orientation tolerance (a) tolerance specification of a hole and (b) cylindrical tolerance zone.

The inequalities established for evaluating the tolerance capability about the parallelism of two holes is

$$\varepsilon_x{}^2 + \varepsilon_y{}^2 \leq \left(\frac{t}{d}\right)^2 \tag{4}$$

where $t = 0.01$ and $d = 20$.

3.2.3.3.2   Position tolerance

The position tolerance is used to describe the true location of a machining feature relative to a datum reference or some other features. For cylindrical features (holes and bosses), the position tolerance is the diameter of the tolerance zone within which the axis of the feature must lie, with the center of the tolerance zone being the true position. Figure 15 (a) shows an example of position tolerance zone for a hole, and Figure 15 (b) defines a cylindrical tolerance zone within which the axis of the hole must lie. However, this cylindrical tolerance zone is different from the one defined above for the orientation tolerance (Figure 15). The difference is that not only the translation but also the rotation can affect the position deviation. For example, the position deviation in the x-axis (Figure 15 (b)) includes the translation and the deviation converted from the rotation. The amount due to the translation displacement is δ_x .The orientation displacement ε_y indicates the slope of the axis, and the net amount of position deviation due to orientation displacement is (h/2) ε_y.   Note that these two factors may not necessarily act in the same direction. If the two factors act in the same direction, the resultant deviation will be the sum of those two factors, otherwise the difference should be taken. In a similar fashion, the two factors affecting the position deviation in the y direction can be calculated. Overall, the deviations in both directions, x and y, are combined together into the set of inequalities.

Figure 15. Position tolerance (a) tolerance specification and (b) cylindrical tolerance zone.

The inequalities established for evaluating the positioning tolerance of a hole is

$$(\delta_x + \frac{h}{2}\varepsilon_y)^2 + (\delta_y + \frac{h}{2}\varepsilon_x)^2 \leq (\frac{t}{2})^2 \tag{5}$$

$$(\delta_x + \frac{h}{2}\varepsilon_y)^2 + (\delta_y - \frac{h}{2}\varepsilon_x)^2 \leq (\frac{t}{2})^2 \tag{6}$$

$$(\delta_x - \frac{h}{2}\varepsilon_y)^2 + (\delta_y + \frac{h}{2}\varepsilon_x)^2 \leq (\frac{t}{2})^2 \tag{7}$$

$$(\delta_x - \frac{h}{2}\varepsilon_y)^2 + (\delta_y - \frac{h}{2}\varepsilon_x)^2 \leq (\frac{t}{2})^2 \tag{8}$$

where t = 0.01 and h = 50.

For non-cylindrical features (slots, pockets, etc.) the position tolerance is the total width of the tolerance zone within which the center plane of the feature must lie, with the center plane of the zone being the true position. Figure 16 shows an example of position tolerance zone for a slot. Similar to the cylindrical case, there are two different factors contributing to the total position deviation, i.e. the position and orientation displacements. The amount of the position displacement is δ_x. For orientation displacement, two rotations need to be considered, about the y-axis and z-axis. Combining all these displacements, the total deviation volume is represented by a set of inequities.
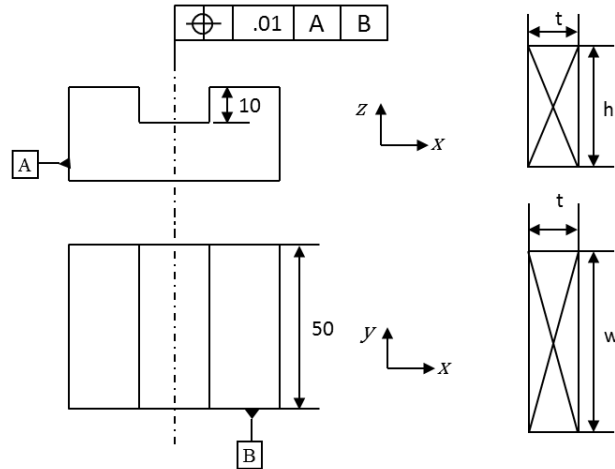
Figure 16. Position tolerance for a slot (a) tolerance specification and (b) tolerance zone.

The inequalities established for evaluating the position tolerance of a slot is

$$-\frac{t}{2} \leq \delta_x + \frac{w}{2}\varepsilon_z + \frac{h}{2}\varepsilon_y \leq \frac{t}{2} \tag{9}$$

$$-\frac{t}{2} \leq \delta_x + \frac{w}{2}\varepsilon_z - \frac{h}{2}\varepsilon_y \leq \frac{t}{2} \tag{10}$$

$$-\frac{t}{2} \leq \delta_x - \frac{w}{2}\varepsilon_z + \frac{h}{2}\varepsilon_y \leq \frac{t}{2} \tag{11}$$

$$-\frac{t}{2} \leq \delta_x - \frac{w}{2}\varepsilon_z - \frac{h}{2}\varepsilon_y \leq \frac{t}{2} \tag{12}$$

where t = 0.01, h = 10 and w = 50.

3.2.3.3.3   Dimension tolerance

The dimensional tolerances can also be represented as a tolerance zone. Figure 17 shows an example to evaluate the dimension tolerance of a part. The amount of positional displacement is δz and the contribution from the orientation displacement can be derived as (h/2) εx.

(a)   (b)
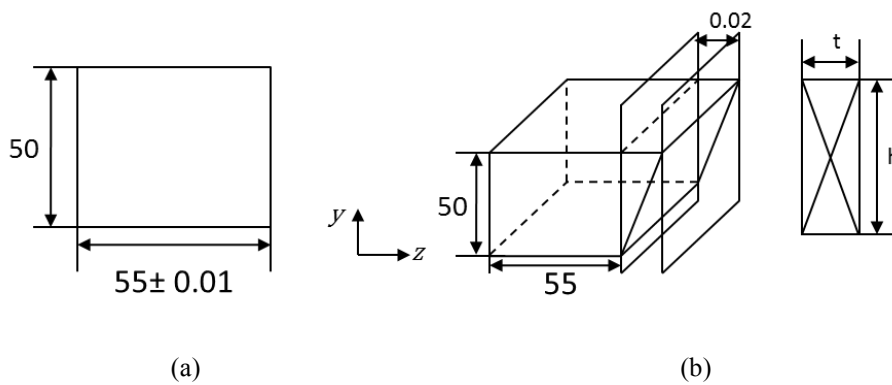
Figure 17. Dimension tolerance (a) position tolerance and (b) deviation volume.

The inequalities established for evaluating the dimension tolerance capability is

$$-\frac{t}{2} \leq \delta_z + \frac{h}{2}\varepsilon_x \leq \frac{t}{2} \qquad (13)$$

$$-\frac{t}{2} \leq \delta_z - \frac{h}{2}\varepsilon_x \leq \frac{t}{2} \qquad (14)$$

where t = 0.02 and h = 50.

In order to satisfy the required GD&T for the machining processes, the inequalities established need to be met. Besides the manufacturing features illustrated above, similar analysis can be done to other manufacturing features such as step, pocket, etc.

3.2.3.4 Constraints and error sources

To evaluate the constraints on the displacements of the machining features with respect to their nominal coordinate system, the error sources of machining processes need to be analyzed. There are different ways to classify the error sources. One way is to classify them into two categories: systematic errors and random errors [12]. Systematic errors are constant and repeatable, whereas random errors arise from random fluctuations in the system. Another way is to classify the errors according to the physical sources.

To find out the physical error sources that affect the accuracy of a manufactured part, it is helpful to review the procedure of manufacturing operations, to investigate potential errors. First, a fixture is set up on a machine table. The locators are then placed on the fixture. Then the workpiece is located and clamped at a proper position in the fixture, so that the workpiece can undergo manufacturing operations. By examining the machining of a component (Figure 18), it can be found that the error sources come from three elements of the machining system: machine tool, cutting tool, fixture. The factors to evaluate the inaccuracies from those sources are listed in Table 15. The errors from different sources will be combined into errors in the nominal feature coordinate system, which are $\delta_x, \delta_y, \delta_z, \varepsilon_x, \varepsilon_y, \varepsilon_z$, and some or all of them will be plugged into the inequalities defined in Section 3.2.3.3, depending on the tolerance specifications and machining features.
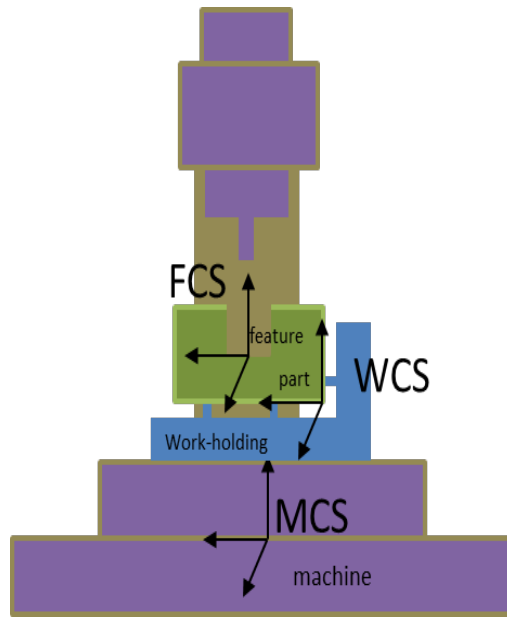
Figure 18. Error sources involved in machining processes.

Table 15. Category of machining error sources

Category	Error parameters
Machine tool	Accuracy
	Repeatability
Cutting tool	Wearing rate
	Tool geometry
Fixture	Position accuracy
	Rotation accuracy

3.2.3.4.1 Machine tool errors

There are different error factors from machine tools, including geometric, thermal, and cutting force. These factors cause resultant volumetric errors at the cutting tool tip, which are the differences between the actual tool tip position and the commanded position. Among them, the major error source is geometric errors [13]. Geometric errors are introduced by the mechanical imperfections of the machine tool structure and the misalignment of machine tool elements. Those elements change gradually due to component wear and tear. The geometric error for a three-axis machine can be measured in terms of 21 parameters [10]. These factors include the linear accuracy in the x-, y-, and z-axes; squareness of the axes to one another; straightness in the

horizontal and vertical direction in each axis; and the rotational errors (roll, pitch and yaw) in all three axes. Due to the unavailability of those data, the machine tool accuracy provided by the machine tool vendor is defined in the manufacturing process library and will be used for calculating the position displacement in x, y, or z direction in machine tool coordinate system.

3.2.3.4.2 Cutting tool errors

For machining processes, the cutting tool gradually wears out, until a new cutting tool has to replace the old one. As the tool wears progressively, the amount of wear will be introduced as dimensioning and geometric errors for the part being produced. Most models use an initial high wearing rate period followed by a steady wearing rate period and a subsequent accelerating wearing rate result in tool failure (Figure 19).
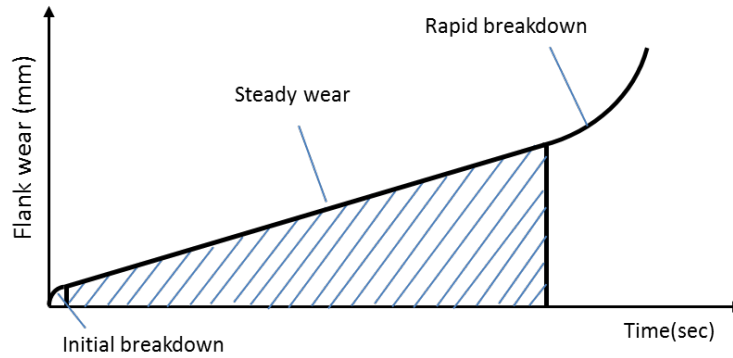
Figure 19. Progressions of the tool wear.

Taylor's equation has been used to evaluate the tool life. The tool life expectancy shown in equation (15) has been used in various research efforts, and is defined as

$$v = \frac{C}{T^n}.$$

(15)

From equation (15), the tool life for any cutting speed can be calculated as

$$T = \sqrt[n]{\frac{C}{v}}$$

(16)

where T is the tool life expectancy, v is the cutting speed, n is Taylor's tool life exponent, and C is the cutting speed when the tool life expectancy is 1 minute. The values of both n and C depend mainly upon the cutting tool and workpiece materials and the cutting environment (cutting fluid application). The value of C depends also on the limiting value of flank wear undertaken (i.e., 0.012 inch, 0.016 inch, 0.060 inch, etc.).

Table 16. Typical values of n and C in Taylor tool life equation

Tooling and workpiece material	n	C (ft./min)
High speed steel cutting tool:		
Non-steel workpiece	0.125	350
Steel workpiece	0.125	200
Cemented carbide cutting tool:		
Non-steel workpiece	0.25	2700
Steel workpiece	0.25	1500
Ceramic cutting tool:		
Steel workpiece	0.6	10000

Typical values of n and C in Taylor tool file equation is shown in Table 16, which are obtained from [7]. The tool reaches its useful life when the tool flank wears to the maximum allowable extent. The maximum allowable wear for cutting tools in various operations is listed in Table 17 [8]. Assuming that the tool is operating in the steady wear phase, the wear rate of the tool of each operation required to manufacture the part can be estimated as,

$$k = \frac{Maximum\_allowable\_wear}{T} \qquad (17)$$

where Maximum_allowable_wear is the maximum allowable wear at which the tool is considered to be at the end of its useful life, T is the tool life calculated from equation (16), k is the tool wear coefficient.

Table 17. Allowable average wear for cutting tools in various operations

Operation	Allowable wear (inch)	
	High-speed steel cutting tool	Carbide cutting tool
Turning	0.060	0.016
Face milling	0.060	0.016
End milling	0.012	0.012
Drilling	0.016	0.016
Reaming	0.006	0.006

With the tool wear coefficient estimated, the amount of wear for each operation can be calculated.

The magnitude of tool wear is $w = kT_p$, where k is the coefficient charactering the rate of tool wear over time, calculated from equation (17), and T_P is the machining operation time. The operation time can be estimated as $T_p = \dfrac{W}{MRR}$, where MRR is the material remove rate, and W is the volume of the material to be removed. The amount of wear calculated is the position displacement in z-axis of the cutting tool coordinate system.

For periphery milling operation, the cutting tool tolerance defined in the manufacturing process library is used for calculating the displacement in x or y axes of the cutting tool coordinates system.

3.2.3.4.3   Fixture errors

Before machining can take place, the workpiece must be held stationary on the machine table using fixtures. This is done by mounting the fixture on the machine table, placing the workpiece on the fixture, and then applying clamping force to hold the workpiece still. Perfect shapes are not achievable because of the errors in the workpiece from raw stock or previous manufacturing processes. From a raw stock to the final product, intermediate workpieces are produced, which will have both dimensional errors and surface roughness that will affect its position and orientation of the workpiece coordinate system Figure 20.

Ideal datum

Real datum
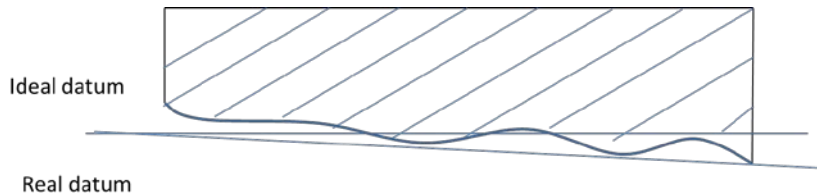
Figure 20. Inaccuracy workpiece bring locating error.

When placing the workpiece on the fixture, because of the workpiece surface imperfection, the workpiece coordinate system (WCS) will not coincide with the locating coordinate system (LCS). Figure 21 shows the effect of machining error coming from fixture. The fixturing error within a single setup is represented as a six tuple, F = $\{\Delta x, \Delta y, \Delta z, \theta x, \theta y, \theta z\}$, which includes the positioning accuracy and rotational accuracy in x, y, and z-axes. The value of the fixturing error depends on the accuracy of the fixture device and the quality of the workpiece surface, which is stored as historical data in the manufacturing process library database.

Approved for public release; distribution unlimited.
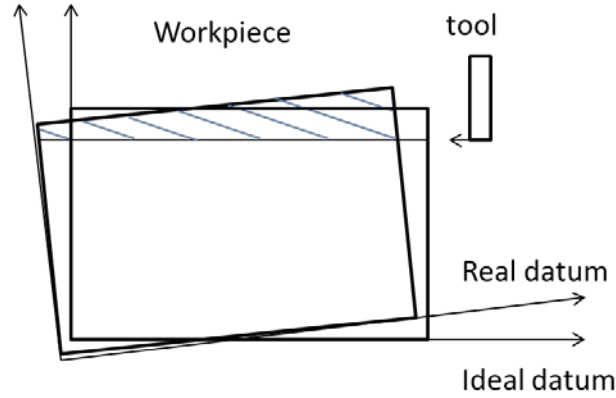
Figure 21. Machining error from fixturing.

3.2.3.4.4 Error integration

As shown in Figure 22 in the nominal feature coordinate system, the value for each entry of the displacement vector: $T = \{\delta_x, \delta_y, \delta_z, \varepsilon_x, \varepsilon_y, \varepsilon_z\}$ is a function of all the errors, including position and orientation errors, from different sources.
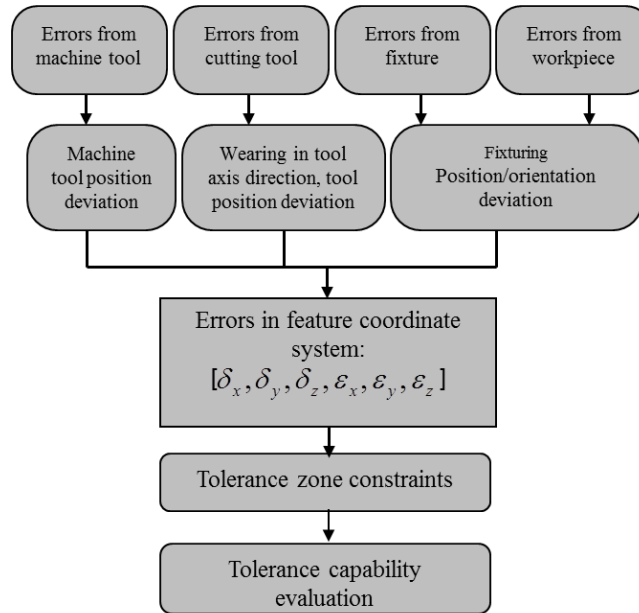
Figure 22. Error integration from different error sources.

For instance, δ_x in the ideal feature coordinate system is the summation of translational errors from machine tool, cutting tool, and fixture. The other values of the displacements in nominal feature coordinate system can be evaluated in the same way. When all the displacements are evaluated, those values will be plugged into the constraints set up in Section 3.2.3.3. The error sources with their IDs are presented in Table 18. Equations (18) and (19) show how the displacements in nominal feature coordinate system are calculated from all the error sources.

Table 18. Error sources and their IDs.

Error source	Machine tool	Cutting tool	Fixture
Source ID	1	2	3

$$\delta_i = \sum_j \delta_j, i = x, y, z; j = 1,2,3 \qquad (18)$$

$$\varepsilon_i = \sum_j \varepsilon_j, i = x, y, z; j = 1,2,3 \qquad (19)$$

3.2.3.5 An illustrative example

A test part pictured in Figure 23 is used to show how tolerance can be evaluated. The inputs for tolerance evaluation are listed below:

Machine tool accuracy (Fryer 5x - 45): 0.0002″.

Cutting tool accuracy (Melin GMGS-404 carbide end mill): 0.002″.

Fixture accuracy (Olympia tool 38-604 4-in bench vise): positioning accuracy 0.0001″, and rotational accuracy 0.0002º.

Raw stock (steel workpiece): 22″ × 11″ × 2.2″.

Cutting tool diameter: 0.5″.

The process parameters such as spindle speed, feed rate were determined based on the following rationale. Depth of cut is considered a user input, with a default value of half of the tool diameter.

$$\text{Spindle speed} = \frac{12}{\pi} \times \frac{SFM}{D} \approx 800 \quad \text{rpm} \qquad (20)$$

where $\frac{12}{\pi}$ is a conversion factor; surface feet per minute SFM = 100 ft/min for mild steel; tool diameter D = 0.5″.

$$\text{Feed rate } f = \text{Spindle speed} \times \text{Chip load per tooth} \times \text{Number of flutes} = 16 \text{ ipm} \quad (21)$$

where spindle speed = 800 rpm; chip load per tooth = 0.005″, which is provided by the manufacturer and available in the library; number of flutes (teeth) = 4, which is physical property of the tool and available in the library.

Using the formulas of spindle speed and feed rate allows us to calculate the material removal rate (MRR).

$$\text{MRR} = \text{Feed Rate} \times \text{Width of Cut} \times \text{Depth of Cut} = 1.5 \text{ in}^3/\text{min} \qquad (22)$$

where width of cut is considered a user input, with a default value of 0.75*D; depth of cut is also considered a user input, with a default value of 0.5*D.

Based on the inputs, the tool life can be calculated as

$$T = \sqrt[n]{\frac{C}{v}} = 256 \text{ min} \tag{23}$$

where C = 200 ft/min for steel, v = 100 ft/min, n = 0.125.

According to Table 17, for end milling operation, Carbide cutting tool will be changed when its wear reaches 0.016 inch. The tool wearing rate can be estimated as

$$k = \frac{Maximum\_allowable\_wear}{T} = 0.625 \times 10\text{-}4 \text{ inch/min} \tag{24}$$

Approved for public release; distribution unlimited.
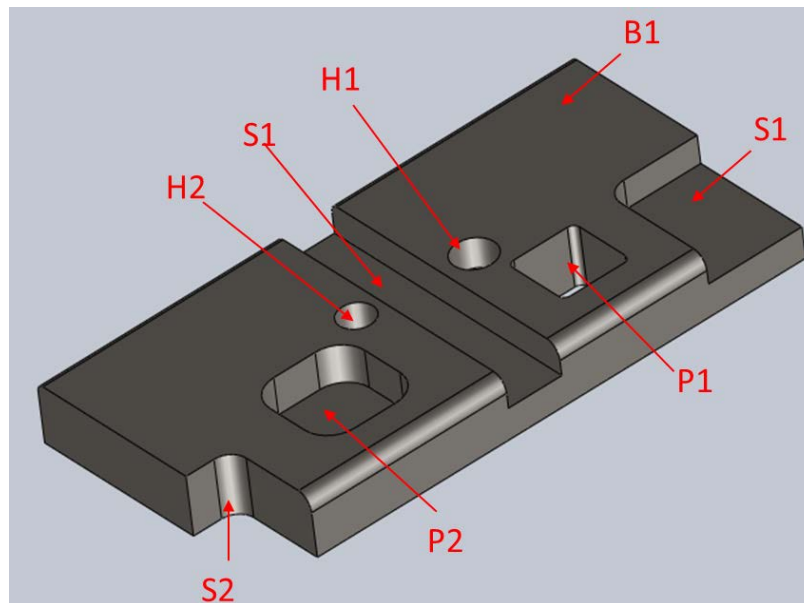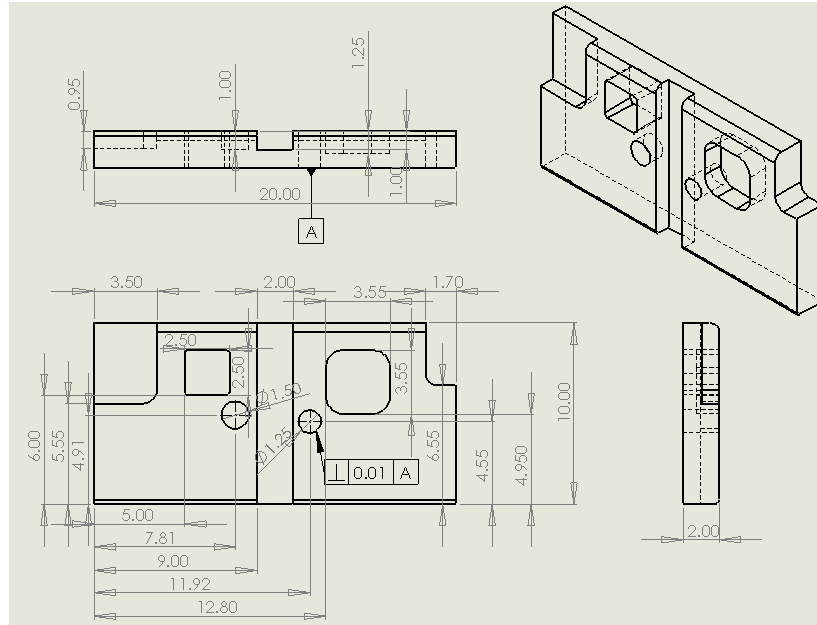
Figure 23. Sample component 2D and 3D drawing.

Four types of features are included in the test part, which are planar surface, slot, step and hole. Eight machining features are to be produced, and the manufacturing sequence is shown in Figure 24. Table 19 shows the detailed manufacturing process plan.
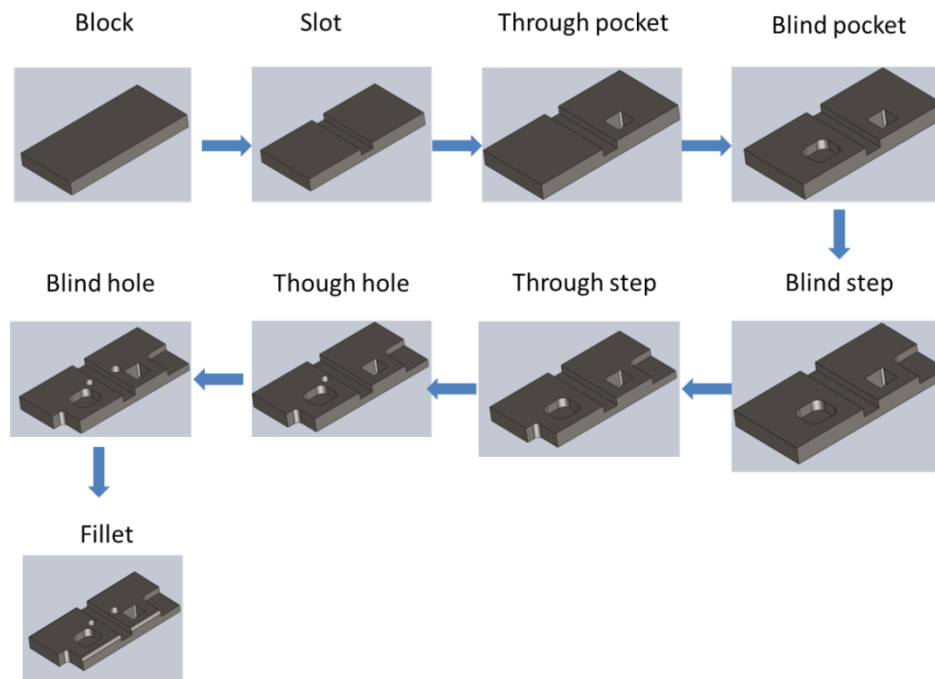
Figure 24. Manufacturing features and machining sequence.

Approved for public release; distribution unlimited.

Table 19. Process plan for the test part.

Process No	Machine	Cutting tool	Feature	Process statement	Dimension 1 (Inch)	Dimension 2 (Inch)	Dimension 3 (Inch)
10	FRYER 5X - 45	Carbide end mill	Block (B1)	Rough milling	20 (±0.01)	10 (±0.01)	2 (±0.01)
20	FRYER 5X - 45	Carbide end mill	Slot (S1)	Rough milling	2 (±0.01)	1 (±0.01)	
30	FRYER 5X - 45	Carbide end mill	Through Pocket (P1)	Rough milling	2.5 (±0.01)	2.5 (±0.01)	
40	FRYER 5X - 45	Carbide end mill	Blind Pocket (P2)	Rough milling	3.55(±0.01)	3.55(±0.01)	1.25(±0.01)
50	FRYER 5X - 45	Carbide end mill	Blind Step (S1)	Rough milling	4.45(±0.01)	3.5(±0.01)	0.95(±0.01)
60	FRYER 5X - 45	Carbide end mill	Through Step (S2)	Rough milling	3.45(±0.01)	1.7(±0.01)	
70	FRYER 5X - 45	Carbide drill	Blind Hole (H1)	Drilling	1.5(±0.01)	1.0(±0.01)	
80	FRYER 5X - 45	Carbide drill	Though Hole (H2)	Drilling	1.25(±0.01)		

Figure 25 shows how the workpiece is set up on the machine table. The first operation is to machine the three planar surfaces, P1, P2, and P3. The time to machine P1 is approximately

$$T_p = \frac{W}{MRR} = 32.3 \, \text{min},$$ where W = 48.4 in³, which is the amount of material to be removed, and MRR = 1.5in³/min, which is the material removal rate. The cutting tool wear during this milling operation is w = kTp = 2.02 ×10-3 inches.

The detailed calculation for evaluating the dimension tolerance of the 2 inches by milling P1 is shown below.
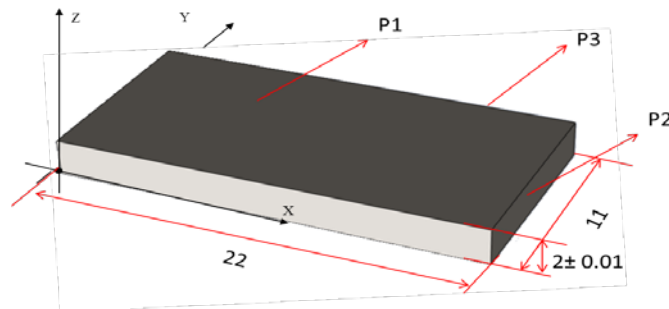
Figure 25. Raw workpiece set up in machine tool coordinate system.

Step 1: set up the feature coordinate system. The normal vector of the planar surface can be assumed to be aligned with the z axis of the feature coordinate system without loss of generality. And the x and y axis are set up in a way such that a right handed coordinate system is formed (Figure 25).

Step 2: analysis of tolerance zone (Figure 26). The tolerance zone is the area between two parallel planes with distance 0.02 inches apart from each other. The locations of the two parallel planes are with respect to the datum plane, which is the bottom plane opposite to P1 as shown by Figure 25.
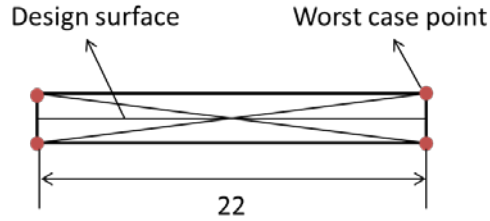
Figure 26. Tolerance zone.

By analyzing the tolerance zone, the following relationship should hold for the design tolerance to be satisfied.

$$-\frac{0.01}{2} \leq \delta_z + \frac{22}{2}\varepsilon_y \leq \frac{0.01}{2}$$

(25)

$$-\frac{0.01}{2} \leq \delta_z - \frac{22}{2}\varepsilon_y \leq \frac{0.01}{2}$$

(26)

Step 3: evaluate the tolerance. δz is the sum of the machine tool accuracy, fixture positioning accuracy, and cutting tool wear. And εy is fixture rotational accuracy in y direction. Worst case scenario is adopted for calculating δz.

$$\delta_z = 0.0002 + 0.0001 + 0.00202$$

(27)

$$\varepsilon_y = 0.0002$$

(28)

Plug equation (27), (28) into equation (25), (26), and the inequalities hold. Therefore, the design tolerance can be satisfied for dimension 3 of the Block. Similarly, the other dimension tolerances listed in Table 19 can be evaluated. We also need to evaluate the perpendicularity tolerance for the though hole (H2). The detail calculation for evaluating the perpendicularity of H2 is also shown below.

Step 1: set up the feature coordinate system (Figure 27). The ideal axis vector of the cylindrical surface can be assumed to be aligned with the z-axis of the feature coordinate system without

loss of generality. And the x and y axis are set up in a way such that a right handed feature coordinate system is formed.
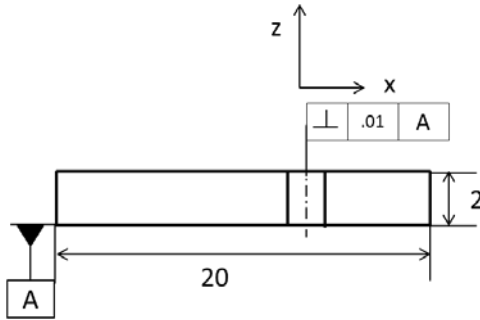
Figure 27. Feature coordinate system.

Step 2: analysis of tolerance zone (Figure 28). The tolerance zone is a cylinder with diameter 0.01 inch.
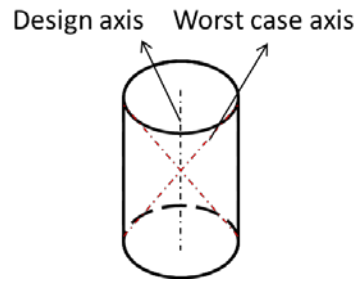
Figure 28. Tolerance zone.

By analyzing the tolerance zone, the following relationship should hold for the design tolerance to be satisfied.

$$\varepsilon_x{}^2 + \varepsilon_y{}^2 \leq \left(\frac{0.01}{1.25}\right)^2 \tag{29}$$

Step 3: evaluate the tolerance. εx and εy are the fixturing rotational accuracy in x and y direction, respectively.

$$\varepsilon_x = \varepsilon_y = 0.0002 \tag{30}$$

Plug equation (30) into equation (29), and the inequality holds. Therefore, the perpendicularity tolerance can be satisfied for H2.

It has been verified from the preceding example that the feature based tolerance evaluation scheme can be extensively used for evaluating the various tolerances in a systematic manner.

3.3   Characterizing Factory and Manufacturing Space

This section details the research efforts to characterize the factory as a component of the library. Additionally, manufacturing and assembly processes are analyzed and captured with respect to the human element of the manufacturing domain.

3.3.1   Introduction

Manufacturing facilities can be broadly defined as buildings where men, material, and machines come together for manufacturing a product or providing a service [1]. Typically, a factory can be partitioned into five hierarchical levels, i.e. facility (factory), shop (area), cell, workstation and equipment (see Figure 29) according to the collaborative work done within ISO and NIST (National Institute of Standards and Technology). This hierarchical organization enables the manufacturing model to provide adequate strategic information at the factory level and operational manufacturing information at the lower shop, cell and station levels. This enables the manufacturing model to provide adequate manufacturing information to the different data-driven applications, i.e. design for manufacture, manufacturing information generation and production planning and control.
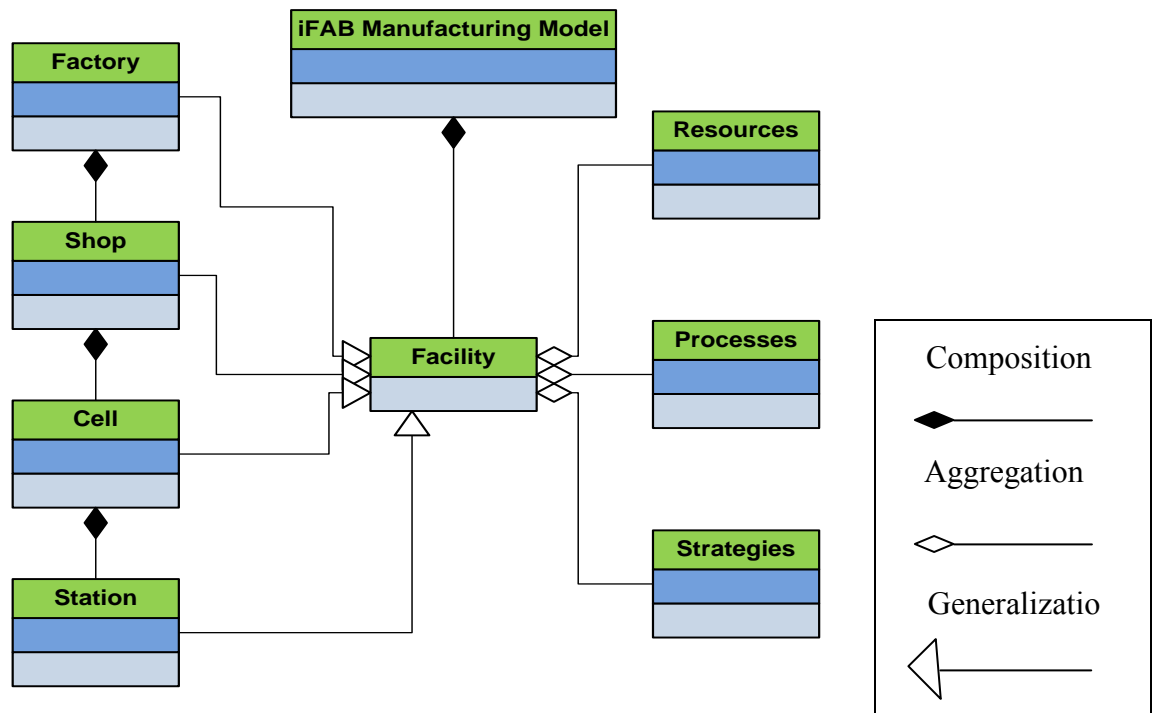
Figure 29. High-level Class Diagram of Facility in iFAB Manufacturing Model

It is important, therefore, to be able to model the facility from various perspectives and at different levels of abstraction in order to provide manufacturing information to meet the needs of all users (e.g., decision makers, engineers, etc.). For example, a manufacturing engineer has a different perception of the capability of a facility than the manager of a shop floor. The information usage is dependent on the activity performed by individual employees in the company, e.g. design for manufacture, generation of process plans and NC codes or the

Approved for public release; distribution unlimited.

formulation and execution of production schedules. Additionally, it must support both generic and specific representations in particular manufacturing enterprises. Finally, it must be able to capture the manufacturing information at different levels of functionality, i.e. from different perspectives.

In addition, there is a need to represent a company's strategic decisions and operational rules, here called manufacturing strategies. Manufacturing strategies represent how a company's resources and processes are organized, composed and deployed to support the realization of the manufacturing function in order to achieve the manufacturing objectives of the company. Therefore, they also impose constraints on the use of a certain type of resource or process. For example, two companies can have the same type of technology, i.e. manufacturing resources and processes, but their manufacturing facilities could be achieving different levels of performance because of each company's decisions on how to apply those resources and processes. This aspect requires a third class of information, i.e. manufacturing strategies, if the manufacturing model is to have the capacity truly to represent the manufacturing capability of a company. Inclusion of the strategies in the manufacturing model enables the representation of the company's specific capability information

Figure 30 illustrates the relationships between facility and other entities in iFAB schema using an ontology model. The elaboration of resources class is shown in Figure 31. . The developed facility schema contains two parts, the facility itself and the material handling equipment. The details of facility schema are shown in Table 20. This schema serves as a generic schema for facilities, which cover factory, shop, cell, and station. The attribute *Type* (1.5) under attribute *Generate* specifies which facility (factory, shop, cell, or station) it applies to and the attribute *Number of subcomponents* specifies the next level component (Shop is subcomponent of factory, cell is subcomponent of shop, station is subcomponent of cell) that it has. The *Composition and Configuration* attribute specifies all the resources (machines, production tools, and material handling equipment) included in a manufacturing facility and their configuration. Some composite attributes are shared by multiple schemas and attributes and are defined reusable components as summarized in Table 21. The *Civil, Architecture, Structural and Geotechnical, Mechanical*, and *Electrical System* aspects are further elaborated by additional attributes. The *Civil* attribute captures field location, all natural objects, and surface improvements aspects of a facility. The *Architecture* attribute captures the floor space, building elements, building support spaces, and miscellaneous architectural element aspects of a facility. The *Structural and Geotechnical* attribute captures the foundations, structural system, equipment supports, and architectural-structural interaction aspects of a facility. The *Electrical System* attribute captures the power, power distribution, illumination, branch circuit work, emergency power, telecommunications and signaling, and other electrical systems. The *Mechanical System* architecture captures the plumbing, HVAC, and compressed air distribution system aspects of a facility

Approved for public release; distribution unlimited.
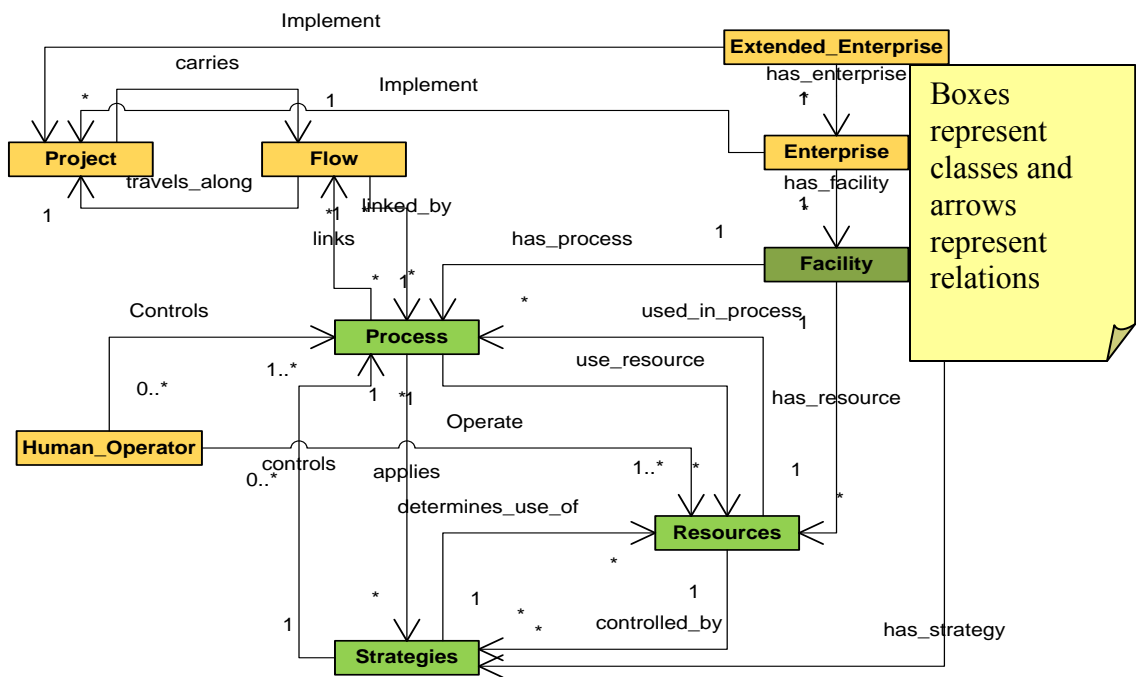
Figure 30. Facility Schema Context Described in an Ontology Model
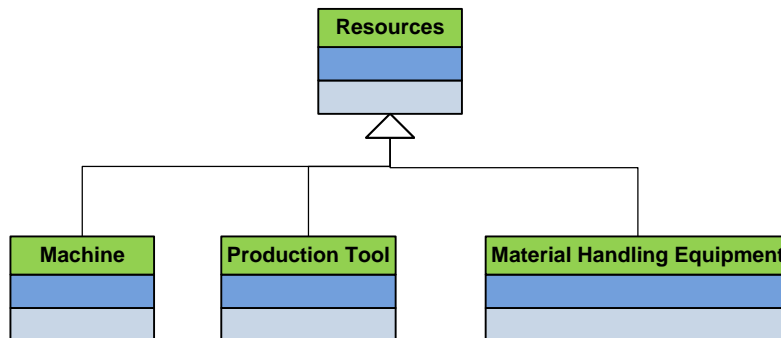
Figure 31. Elaboration of Resources Class

Approved for public release; distribution unlimited.

3.3.2 Development of Facility Schema

Table 20. Facility schema

1	General	1.1	ID		
		1.2	Name		
		1.3	Function		
		1.4	Description		
		1.5	Type		
		1.6	Environmental Requirements		
2	Management Info	2.1	Maintenance		
		2.2	Lead Time		
		2.3	Open Time		
		2.4	Construction/Acquisition/installation Time		
		2.5	Setup Time		
		2.6	Removal Time		
		2.7	Availability		
3	Economic Info	3.1	Investment Cost		
		3.2	Depreciation Period		
		3.3	Operation Cost		
4	Composition and Configuration	4.1	Machines		
		4.2	Machine Number and Variance		
		4.3	Production Tools		
		4.4	Production Tool Number and Variance		
		4.5	Material Handling Equipment		
		4.6	Material Handling Equipment Number and Variance		
		4.7	Configuration		
		4.8	Number of subcomponents		
		4.9	Subcomponent layout		
		4.10	Strategies		
5	Civil	5.1	General	5.1.1	Conforming Standards
				5.1.2	Building Orientation
				5.1.3	Traffic
				5.1.4	Site Lighting
				5.1.5	Site Signage
				5.1.6	Pedestrian and Vehicular Circulation
		5.2	Site	5.2.1	Location
				5.2.2	Land Ownership
				5.2.3	Topography
				5.2.4	Site Physical Characters
				5.2.5	Undesirable Conditions Surrounding the Site
				5.2.6	Permanent site features
				5.2.7	Climate
				5.2.8	Noise
				5.2.9	Wind Conditions
		5.3	Landscape	5.3.1	Planting
6	Architecture	6.1	General	6.1.1	Area

				6.1.2	Floor layout
				6.1.3	Building/space function
				6.1.4	Building Components
				6.1.5	Conforming Standards
		6.2	Economic Info		
		6.3	Acoustics	6.3.1	Ambient Noise Level
				6.3.2	Noise Isolation
				6.3.3	Noise Isolation Class
				6.3.4	Reverberation Control
				6.3.5	Building Acoustics Class
		6.4	Exterior	6.4.1	Configuration and Orientation
				6.4.2	Roofing
				6.4.3	Support of Roof-Mounted Equipment
				6.4.4	Windows and Glazing
				6.4.5	Building Entry
		6.5	Interior	6.5.1	Floor
				6.5.2	Ceiling
				6.5.3	Door
				6.5.4	Finishes
		6.6	Building Support Space	6.6.1	Service Area Function
				6.6.2	Service Area Physical Properties
		6.7	Building Accessories		
7	Structural and Geotechnical	7.1	General	7.1.1	Structure Type
				7.1.2	Conforming Standards
				7.1.3	Loads and Stress
				7.1.4	Protection Level
		7.2	Economic Info		
		7.3	Building Elements	7.3.1	Building Materials
				7.3.2	Exterior Wall
				7.3.3	Exterior Window
				7.3.4	Interior Window
				7.3.5	Non Window Opening
				7.3.6	Loading dock
				7.3.7	Venting
		7.4	Foundation	7.4.1	Depths
				7.4.2	Soil bearing capacity
				7.4.3	Differential settlements
				7.4.4	Protection and Support of Adjoining Property
				7.4.5	Impacts on adjoining
				7.4.6	property
		7.5	Structural System	7.5.1	Stability
				7.5.2	Factor of Safety
				7.5.3	Strength
		7.6	Equipment Supports	7.6.1	Vibration
				7.6.2	Design Loads
				7.6.3	Vibration Isolation
				7.6.4	Equipment Anchor
		7.7	Architectural- Structural Interaction	7.7.1	Drift
				7.7.2	Waterproofing
				7.7.3	Cladding and Insulation
8	Mechanical	8.1	General	8.1.1	Conforming Standards
		8.2	Economic Info		
		8.3	Acoustics		

		8.4	Accessibility		
		8.5	Plumbing	8.5.1	Floor Drain
				8.5.2	Sanitary System
				8.5.3	Storm Water Drains
				8.5.4	Water Supply
				8.5.5	Gas Piping
		8.6	HVAC	8.6.1	Heat Losses
				8.6.2	Heat Gains
				8.6.3	Air Cleaning
				8.6.4	Temperature Control
				8.6.5	Humidity Control
				8.6.6	Ventilation
				8.6.7	Air Distribution Devices
		8.7	Security		
9	Electrical System	9.1	General	9.1.1	Conforming Standards
				9.1.2	Special Power Requirement
				9.1.3	Vulnerability
				9.1.4	Equipment Room
		9.2	Economic Info		
		9.3	Power	9.3.1	Capacity
				9.3.2	Power Source Type
				9.3.3	Load type
				9.3.4	Power Type
				9.3.5	Voltage
				9.3.6	Amperage
				9.3.7	Receptacle Type
				9.3.8	Grounding
				9.3.9	Frequency
				9.3.10	Wave Form
		9.4	Emergency Power		
		9.5	Illumination	9.5.1	Lighting Requirement
				9.5.2	Luminaries
				9.5.3	Switch
				9.5.4	Grounding
				9.5.5	Lighting Protection
				9.5.6	Emergency Lighting
		9.6	Power Distribution	9.6.1	Type
				9.6.2	Power Distribution Equipment
				9.6.3	Raceways
				9.6.4	Insulation
		9.7	Service	9.7.1	Service Characteristics
				9.7.2	Service Conductors
				9.7.3	Metering
				9.7.4	Service Equipment
		9.8	Telecommunications and Signaling System		
		9.9	Branch Circuit Work	9.9.1	Wiring and Capacities
				9.9.2	Switching
				9.9.3	Receptacles

Table 21. Reusable Composite Attributes

	Economic_Info			Vibration
1	Investment_cost		1	Frequency
2	Depreciation_period		2	Strength
			3	Wave
	Structure			
1	Shape and Dimension			**Availability**
2	Material		1	Monthly availability
3	Finishing		2	Reliability
4	Foundation		3	Failure Interval (type,mean,std dev)
5	Anchor		4	Repair Interval(type,mean,std dev)
6	Special requirements		5	Mean time to failure
	Environment Requirements			**Maintenance**
1	Temperature		1	Maintenance Req
2	Humility		2	Repairing history
3	Ventilation		3	Part replacement history
4	Lighting		4	Maintenance - Previous service date
5	Anti Dust		5	Maintenance - Next service date
6	Anti Static		6	Maintenance Procedure
7	Noise		7	Maintenance Schedule
8	Vibration			
9	Other specific requirements			**Accessibility**
			1	Access Space
			2	Access Entrance
			3	Access Method
			4	Clearance
			5	Access Equipment

The floor schema shown in Table 22 captures the space requirements and specifications of all resources in a facility. The *2:Activity* attribute identifies the usage of a piece of floor space, which can be at any level of abstraction. For example, at the highest level, the *2:Activity* can take the value of manufacturing, production services, employee service, office, outside areas, etc; within the manufacturing activity, it can be fabrication, paint, assembly, etc.; within the fabrication Activity, it can be strip sheer, punch press, milling, stamping, lathing, etc. Some explanations of the attributes in the floor schema are provided at the right side of the table along with some sample data.

3.3.3   Development of Floor Space Schema

Table 22. Floor Space Schema

			Examples and Description
Floor	1	Usage Description	fabrication area, machine, production services, employee services, office, outside area, walkways …
	2	Activity	Fabrication, welding, paint, assembly and P.O., receiving, Stores, Warehouse, Shipping, Restrooms, Maintenance, Tool room, Locker room, Office, …
	3	Activity Number	Index of Activity
	4	Activity Relations	
	5	Location Specification	Three dimensional coordinate of the lower-left corner at z=0 plain of the space
	6	Location number	Area allocation number in floor layout diagram
	7	Layer	ground level, 1st layer, 2nd layer, …
	8	Placement	overhead, Clear space truss level, truss level, roof, …
	9	Shape and Dimension	
	10	Structure	Concrete, …
	11	Figure	Floor layout diagram for this component
Shape and Dimension	1	length	
	2	Width	
	3	Height	
	4	Shape	
	5	Shape specific dimension	
	6	Area	
	7	Allocated block numbers	(optional) If the floor is divided into small identical-size rectangle blocks with each block indexed. This is the index number of the block
	8	Layout Figure Number	
Block Definition			
Block	1	Block Number	
	2	Length	
	3	Width	
	4	Height	
	5	Location Number	

Aggregate Factors	1	Relationship	100						
	2	Flow Intensity	0						
Reason Code	1	Code	1	2	3	4	5	6	…
	2	Reason	paint fumes	material flow	dust	urgency of contact	noise	share equipment	…
Relation type (Activity Code)	1	A	Absolute necessary that these two departments be next to each other
	2	E	Especially important
	3	I	Important
	4	O	Ordinary closeness

	5	U		Unimportant
	6	X		Closeness undesirable

	Activity Relations
1	Activity
2	Relation type (Activity Code)
3	Relation weights
4	Max Distance From
5	Reason code
6	Flow Intensity
7	Aggregate Value

3.3.3.1 Space Planning

A total plant size and shape is needed very early in the project in order to design the building. Each department's (office, or service area) space needs are analyzed and determined separately. For example, the space might be broken down into manufacturing space, production services space, employee service space, office space, and outside area space. Each of these space requirements can be further broken down and analyzed separately. These sub-space space requirements along with the floor layout determine the upper-level activity's space requirements and so on and so false. The total plant size and shape can thus be obtained from such bottom-up approach.

Once the size of every department, office, and support facility has been determined, space can be allocated to each activity per the dimensionless block diagram's layout, which is the result of the activity relationship chart and the worksheet. Flow analysis can then follows the dimensionless block diagram. These steps are briefly described below and how library data can be used to design floor layout is illustrated with sample data.

One of the first things to be considered in design plant layout is assessing how important a department, office, or service facility to be close to another department, office, or service facility. Such analysis can be conducted using the activity relationship diagrams, also called affinity analysis diagram. It shows the relationship of every department, office, or service area with every other department and area.

A dimensionless block diagram serves as a basis for the master layout and plot plan. A block within the dimensionless block diagram has the activity number placed in the center of the block and its A, E, I , and O relationships placed at four corners and X relationships placed below the activity number.

Flow analysis can then be performed on the dimensionless block diagram, Starting with receiving, show the movement of material to stores, to fabrication, to welding, to paint, to assembly and packout, to warehouse, and to shipping. Flow analysis will ensure that important relationships will be maintained and that the layout makes sense.

With the space requirements and the product flows identified, the building can be divided into departments with activities allocated to a piece of floor space to produce a full layout. The detailed layout with the placement of every piece of equipment can then proceed in a similar manner. Once the area allocation procedure produces a final plan, the architect can start on the building design and construction.

3.3.4 Material Handling Equipment Schema

The material handling schema is shown in Table 23, with populated with four examples of material handling equipment: a bridge crane, a jib crane, a revolver crane, and a fork truck. There are literally thousands of pieces of material handling devices. These pieces of equipment vary from the most basic manual tools to the most sophisticated computer-controlled material handling systems that can incorporate a vast array of other manufacturing and control functions. Almost as varied and numerous are the classification strategies and methods of material handling equipment [21]. Therefore, it is impossible to develop an ad hoc schema for each type of material handling equipment. The developed schema tries to capture the most common attributes shared by most material handling equipment and use the *SpecificAttributes* attribute to capture the type specific attributes. The set of *SpecificAttributes* attributes is determined by the *Type* attribute, which specifies the type of the equipment (e.g., train and railroad track, conveyer system, automated guided vehicles, Jib cranes, bridge cranes, motorized vehicles, pallets, and skids). Note that both the *Classification* and the *Type* attribute are enumerative attributes, the possible values of which are listed in Table 23. These values are identified from the information on the website of the Material Handling Industry of America (MHIA) [22].

3.3.5 Heuristics of Material Handling Equipment Planning

Material handling is the function of moving the right material to the right place in the right time, in the right amount, in sequence, and in the right condition to minimize production cost. Estimates of the cost of MH reach 20-25% of total manufacturing labor cost in the United States according to [22].

3.3.5.1 Goals

The primary goal is to reduce unit costs of production. All other goals are subordinate to this goal. The following subgoals are a good checklist for cost reduction [21];

1. Maintain or improve product quality, reduce damage of materials
2. Promote safety and improve working conditions
3. Promote productivity
   a. Material should flow in a straight line
   b. Material should move as short a distance as possible
   c. Use gravity! It is free power
   d. Move more material at one time
   e. Mechanize material handling
   f. Automate material handling
   g. Maintain or improve material handling / production ratios
   h. Increase throughput by using automatic material handling equipment
4. Promote increased use of facilities

a. Promote the use of building cube
b. Purchase versatile equipment
c. Standardize material handling equipment
d. Maximize production equipment utilization using material handling feeders.
e. Maintain, and replace as needed, all equipment and develop a preventive maintenance program
f. Integrate all material handling equipment into a system
5. Reduce tare weight (dead weight)
6. Control inventory

Table 23. Material Handling Equipment Schema Populated

1.	General		4	Economic		7	Usage Info
1.1	ID		4.1	Initial Cost		7.1	Source Node
1.2	Classification		4.2	Setup Cost		7.2	Source position
1.3	Type		4.3	Annual Operating Cost		7.3	Destination Node
1.4	Used in MH stage		4.4	Annual Maintenance Cost		7.4	Destination position
1.5	Make		4.5	Salvage Value		7.5	Capability
1.6	Model		4.6	Power Consumption		7.6	Mobility
1.7	Floor Space Req. (W x D x H)					7.7	Load
1.8	Supporting Facility Req.		5	Availability		7.8	Load/Unload time
1.9	Life expectancy		5.1	Market Availability		7.8	Material Handled
1.10	Support Structures		5.2	Order Process time		7.10	Material Characteristic
1.11	Attachments &Accessories		5.3	Spare part availability		7.11	Loading manner
1.12	Automation		5.4	Operating hours		7.12	Batch Size
1.13	Power Source		5.5	Monthly availability		7.13	Transportation Rate
1.14	Power Specifications		5.6	Reliability		714	Control System
1.15	Movement		5.7	Failure Interval (type, mean, std dev)		7.15	Consumptions and rate
1.16	Gross weight		5.8	Repair Interval(type, mean, std dev)		7.16	Operation
1.21	Conforming standards		5.9	Part Inventory		7.17	Operation Temperature
			5.10	Mean time to failure		7.18	Status

2	Management
2.1	Installation time
2.2	Setup time
2.3	Removal Time
2.4	Operator Skill Achievement
2.5	Number in use
2.6	Number in Inventory
2.7	Failure Scenario
2.8	Safety Features

6	Motion
6.1	Type
6.2	Speed
6.3	Route
6.3	Route length

8	Accessibility
8.1	Access Space
8.2	Access Entrance
8.3	Access method
8.4	Access equipment
8.5	Clearance

9	Operator Info
9.1	Operator ID

3	Maintenance
3.1	Maintenance Req.
3.2	Repairing history
3.3	Part replacement history
3.4	Maintenance - Previous service date
3.5	Maintenance - Next service date
3.6	Maintenance Procedure
3.7	Maintenance Schedule

9.2	Operator Skill Level
9.3	Operation Time

10. Type Specific Attributes

	Bridge Crane		Jib Crane		Revolver Crane		Fork Lift
10.1	CMAA rating	10.1	CMAA rating	10.1	CMAA rating	10.1	Fork Length
10.2	Number of Cells	10.2	Lifting Mechanism	10.2	Lifting Mechanism	10.2	Tires
10.3	Lifting Mechanism	10.3	Max Horizontal throw	10.3	Span	10.3	Fork Width
10.4	Span	10.4	Max Radius of Gyration				
10.5	Lifting Height	10.5	Max Rotation of Moment				
		10.6	Angle of Revolution				
		10.7	Lifting Speed (Normal/Slow)				
		10.8	Travel Speed				
		10.9	Angle of Revolution Speed				

7.10	Material Characteristic	
7.10.1	Physical state	Solid, liquid, or gas
7.10.2	Size	Volume or length X width X height
7.10.3	Weight	Weight per piece or Weight per unit volume
7.10.4	Shape	Long and flat, round, square, other
7.10.5	Condition	Hot, cold, wet, other
7.10.6	Safety risk and risk of damage	Explosive, flammable, toxic, fragile, other

3.3.5.2 Categorization

Material handling equipment includes:

Transport Equipment: industrial trucks, Automated Guided vehicles (AGVs), monorails, conveyors, cranes and hoists.

Storage Systems: bulk storage, rack systems, shelving and bins, drawer storage, automated storage systems.

Unitizing Equipment: palletizers

Identification and Tracking systems

3.3.5.3 Design Considerations

The materials to be transported by material handling equipment can be broadly categorized as physical state, size, weight, shape, condition, and safety risk and risk of damage. Different category of materials should be measured differently. Figure 32 presents rules of thumb for choosing MHE according to the quantity of material moved and the move distance.
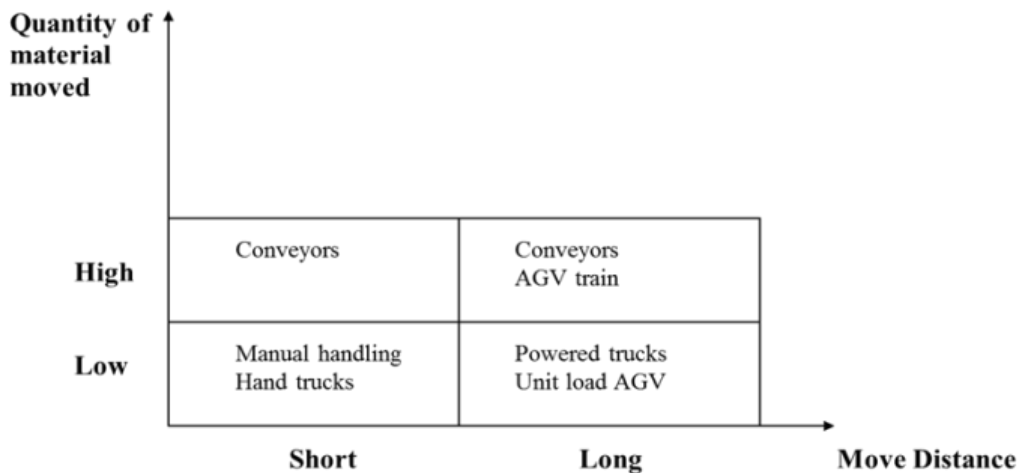
Figure 32. Material Handling Equipment and Flow Quantity and Distance

3.3.5.4 Principles of Material Handing

The College Industry Council on Material Handling Education (CICMHE) of the material Handling Institute, Inc., has adapted the 10 principles of material handing as summarized below (adapted from [21]). These principles could be included in the library as selection strategies, either as library data or as part of library services.

Planning Principle: All material handling should be the result of a deliberate plan where the needs, performance objectives and functional specification of the proposed methods are completely defined at the outset.

Standardization Principle: Material handling methods, equipment, controls and software should be standardized within the limits of achieving overall performance objectives and without sacrificing needed flexibility, modularity and throughput.

Work Principle: Material handling work should be minimized without sacrificing productivity or the level of service required of the operation.

Economic Principle: Human capabilities and limitations must be recognized and respected in the design of material handling tasks and equipment to ensure safe and effective operations.

Unit Load Principle: Unit loads shall be appropriately sized and configured in a way which achieves the material flow and inventory objectives at each stage in the supply chain.

Space Utilization Principle: Effective and efficient use must be made of all available space.

System Principle: Material movement and storage activities should be fully integrated to form a coordinated, operational system which spans receiving, inspection, storage, production, assembly, packaging, unitizing, order selection, shipping, transportation and the handling of returns.

Automation Principle: Material handling operations should be mechanized and/or automated where feasible to improve operational efficiency, increase responsiveness, improve consistency and predictability, decrease operating costs and to eliminate repetitive or potentially unsafe manual labor.

Environmental Principle: Environmental impact and energy consumption should be considered as criteria when designing or selecting alternative equipment and material handling systems.

Life Cycle Cost Principle: A thorough economic analysis should account for the entire life cycle of all material handling equipment and resulting systems.

3.4   Develop Manufacturing Capability and Process Modeling Language (MCPML)

The role of MCPML is to:

- capture the constraints imposed by the fabrication technology, tools, and human capabilities of a foundry, and
- represent the various manufacturing processes, consisting of manufacturing steps, realized by machines and tools, possibly operated by humans performed by a foundry.

The conceptual diagram in the next figure summarizes these points.
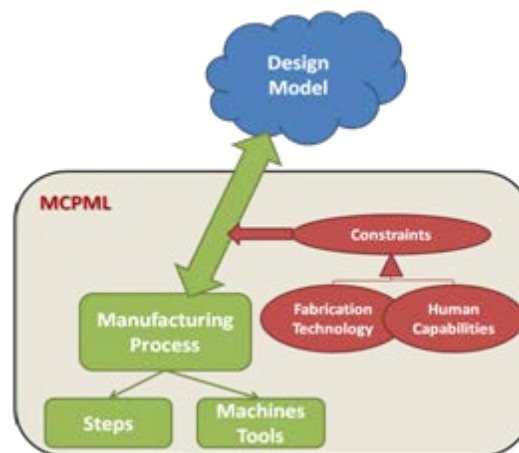
Figure 33. The Scope of MCPML

In order to achieve these objectives, MCPML was created as a novel, domain-specific modeling language that provides specific features to address each objective. The previous section described the various modules that were defined to capture the various entities and attributes that

Approved for public release; distribution unlimited.

make up the manufacturing capabilities of an iFAB foundry. This section provides a more formal basis for the MCPML by defining metamodels for the various library elements. These metamodels capture the entities, relationships, attributes and constraints of the manufacturing capability domain. The library is then implemented as a set of databases (to store the data associated with library entities) and services (to provide the business logic implementing the semantics of the MCPML and to provide other "helper" functions to facilitate access to and use of the library). The remainder of this section describes the capabilities of the language and its metamodel. Appendix C provides background on the metamodeling style followed here.

3.4.1 The MCPML Metamodel

The MCPML metamodel is segmented into several areas, as shown in the picture below. Note that the model database (that is configured based on the model) has a similar segmentation: for each area (i.e. model category) in the metamodel there is a corresponding folder in the model database. The areas are as follows.
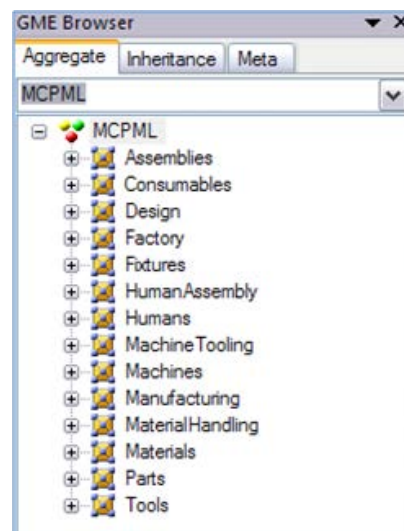
Figure 34. Metamodel Areas

- Design: This area holds design models (or references to CAD models) imported from the META design tools.
- Assemblies: Models for assemblies, i.e. complex, reusable[*] ensembles of parts that are built into a system during a manufacturing process. Example: Diesel engine.
- Parts: Models for parts, i.e. simple, reusable, standardized products that are incorporated into a system during a manufacturing process. Example: Nuts and bolts.
- Consumables: Models for materials that are consumed (and possibly converted into waste products) in the course of manufacturing. Example: Lubricants used in drilling.
- Materials: Models for stock materials that are being worked on during the process to produce parts. Example: Steel plates.

---

[*] Reusability here means that the kind of entity can be used in many, different manufacturing processes.

- Fixtures: Models for devices that can be employed in the process to assist with the assembly or construction. Example: Fixture to hold an engine block while the engine is being assembled.
- Material handling: Models for material handling equipment. Example: Conveyor system.
- Tools: Models for hand-tools that can be used in the process. Example: Torque wrench.
- Machines: Models for manufacturing machines. Example: CNC machine.
- Machine tooling: Models for specialized tools used in the machines. Example: Drill bits.
- Humans: Models for human resources and their skills. Example: John Doe, CNC operator.
- Human assembly: Models for assembly processes where human activities play a major role. Example: Assembling an engine.
- Manufacturing: Models for manufacturing processes that include manufacturing of parts, building assemblies, executing automated and human assembly steps, and integrating the final product.
- Factory: The top-level container for models of available resources in a foundry: assemblies, parts, consumables, materials, fixtures, material handling equipment, hand tools, machines and machine tooling, and humans (with specific skills).
-

Tools are modeled according to the metamodel shown in Figure 35. Tools can be abstract (representing generic tools) and specific (representing unique, special instances of tools). The tool models capture various characteristics and features of the tools that are available in the foundry. The characteristics were selected based on input from manufacturing experts and allow the precise description of the tools, such that given a design an industrial engineer (possibly assisted by some computational tool) can select the right tool for the various steps of the manufacturing process.
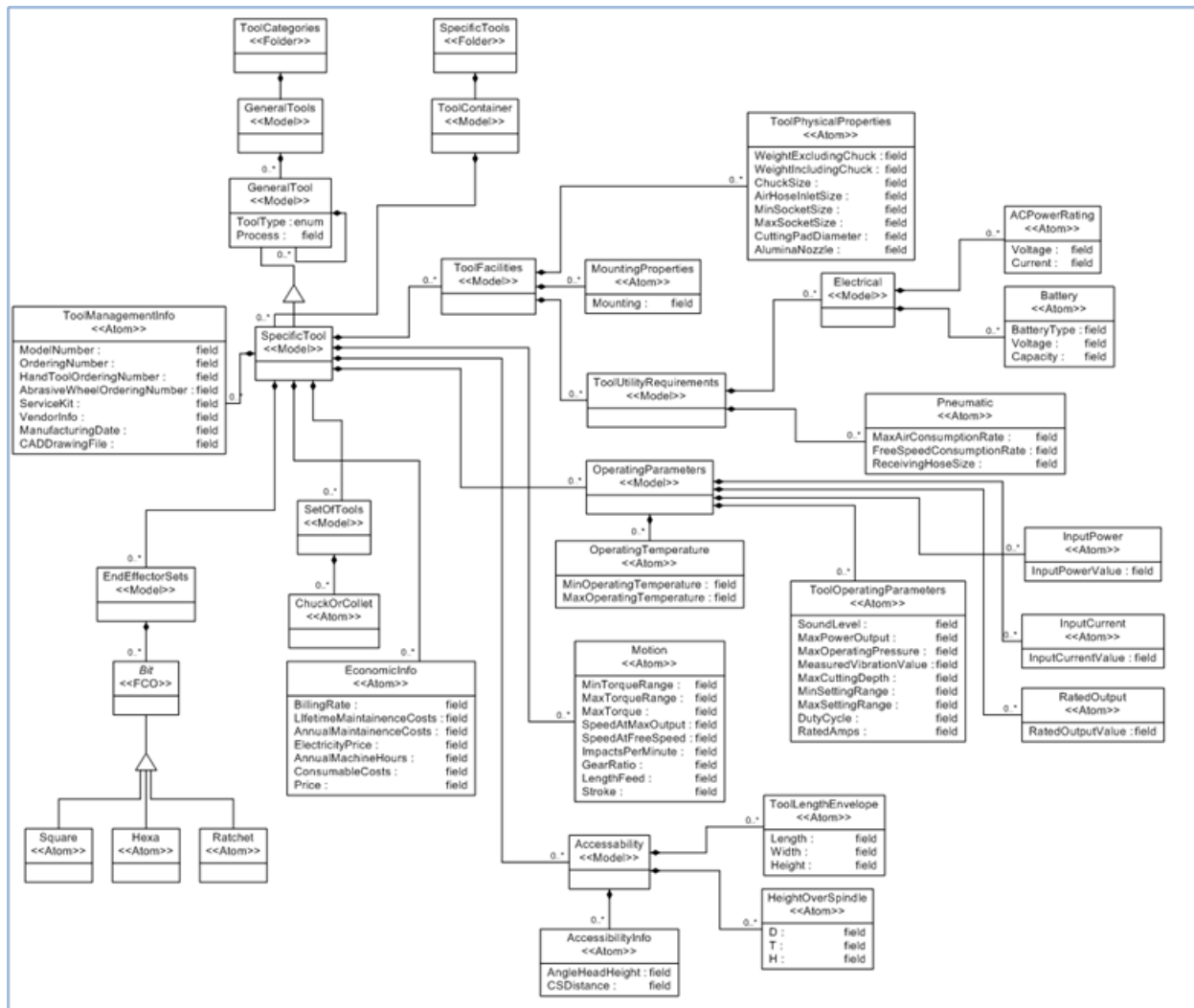
Figure 35. Metamodel for Tools

Models for *Machine Tooling* (metamodel shown in Figure 36) capture the tool 'bits' that are used in the machines (e.g. a CNC machine). These bits have a number of properties that influence what materials the tool can work on, its performance, and other relevant properties (e.g. material properties, economic information, etc.).

Human Assembly models (metamodel shown in Figure 37) capture the types of various human activities during the assembly process. The top-level model is the Generic Assembly Task model that consists of a number of Primitive Tasks (that are contained within a Primitive Assembly Task Library). The Human Assembly Library contains Activities (that contain a set of Hand Postures), and each activity consists of one or more Generic Assembly Tasks. This is the metamodel for HAMML described in Sections 3.1.4 and 4.1, and represented as a spreadsheet in Appendix A. The models that follow the Human Assembly metamodel form a sophisticated library of human capabilities.

Approved for public release; distribution unlimited.
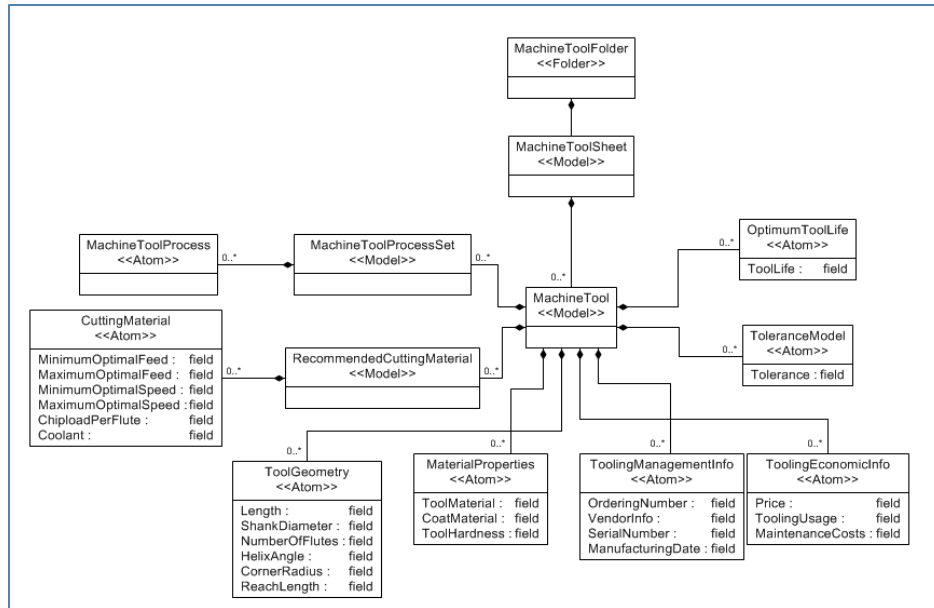
Figure 36. Metamodel for Machine Tooling
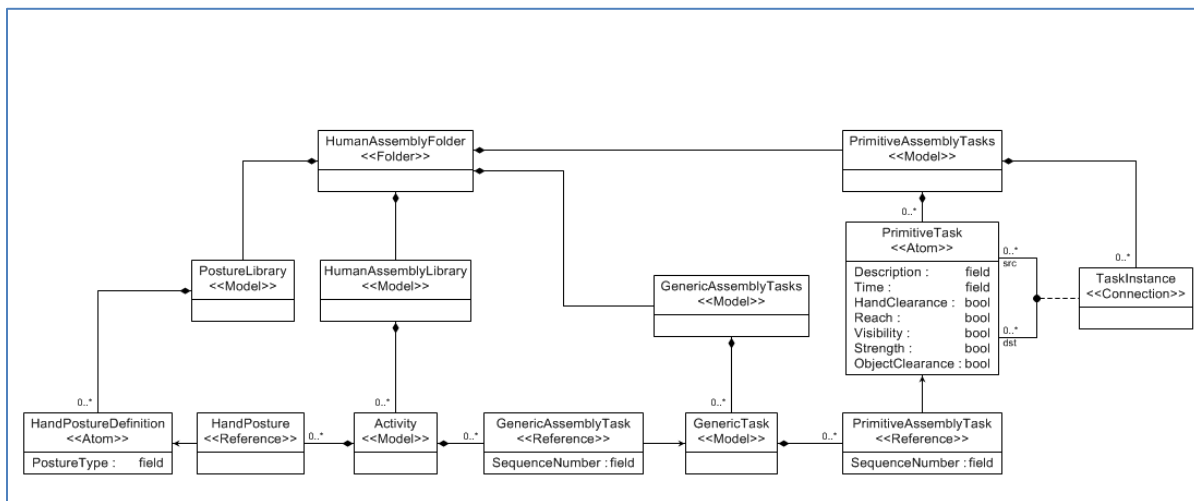
Figure 37. Metamodel for Human Assembly Processes

Material Handling equipment models (see metamodel in Figure 38) provide detailed information about all the MHE in a foundry. The relevant properties include, for instance, geometry, physical load limits, operational speed, economic indicators, and others.

Approved for public release; distribution unlimited.
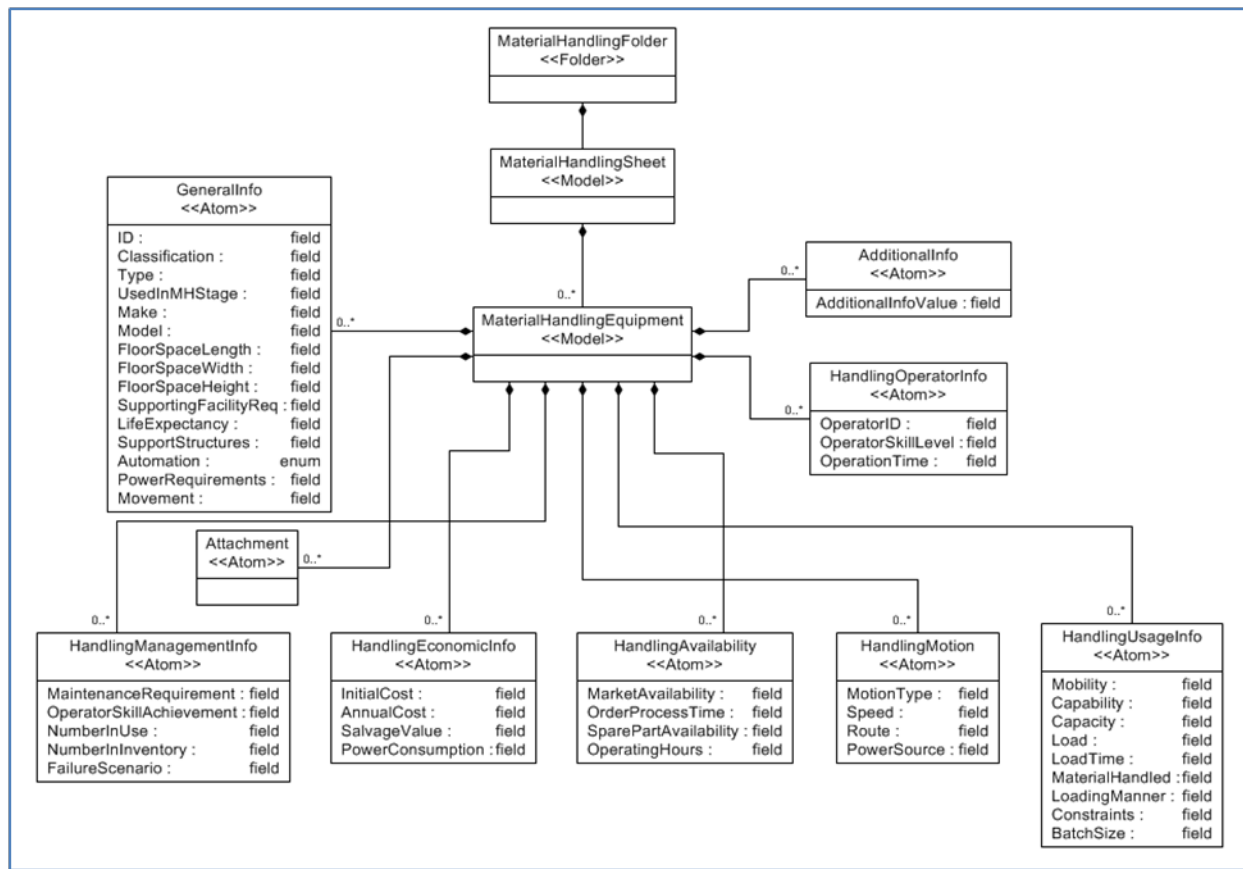
Figure 38. Metamodel for Material Handling

The model categories discussed up to this point describe what is available in a foundry (machines, tools, equipment), and what human assembly steps are feasible (e.g. presumably provided by skilled workers). Some other categories fall into this group as well (assemblies, parts, consumables, materials, fixtures, human resources). For capturing how actually the manufacturing of a design happens, *Manufacturing Process* models are used.

Manufacturing Processes (metamodel shown in Figure 39) are modeled as sequences of steps (where steps are hierarchical, i.e. they can contain other steps, etc). A step can have input and output ports that represent the flow of resources in and out of the step.

Approved for public release; distribution unlimited.
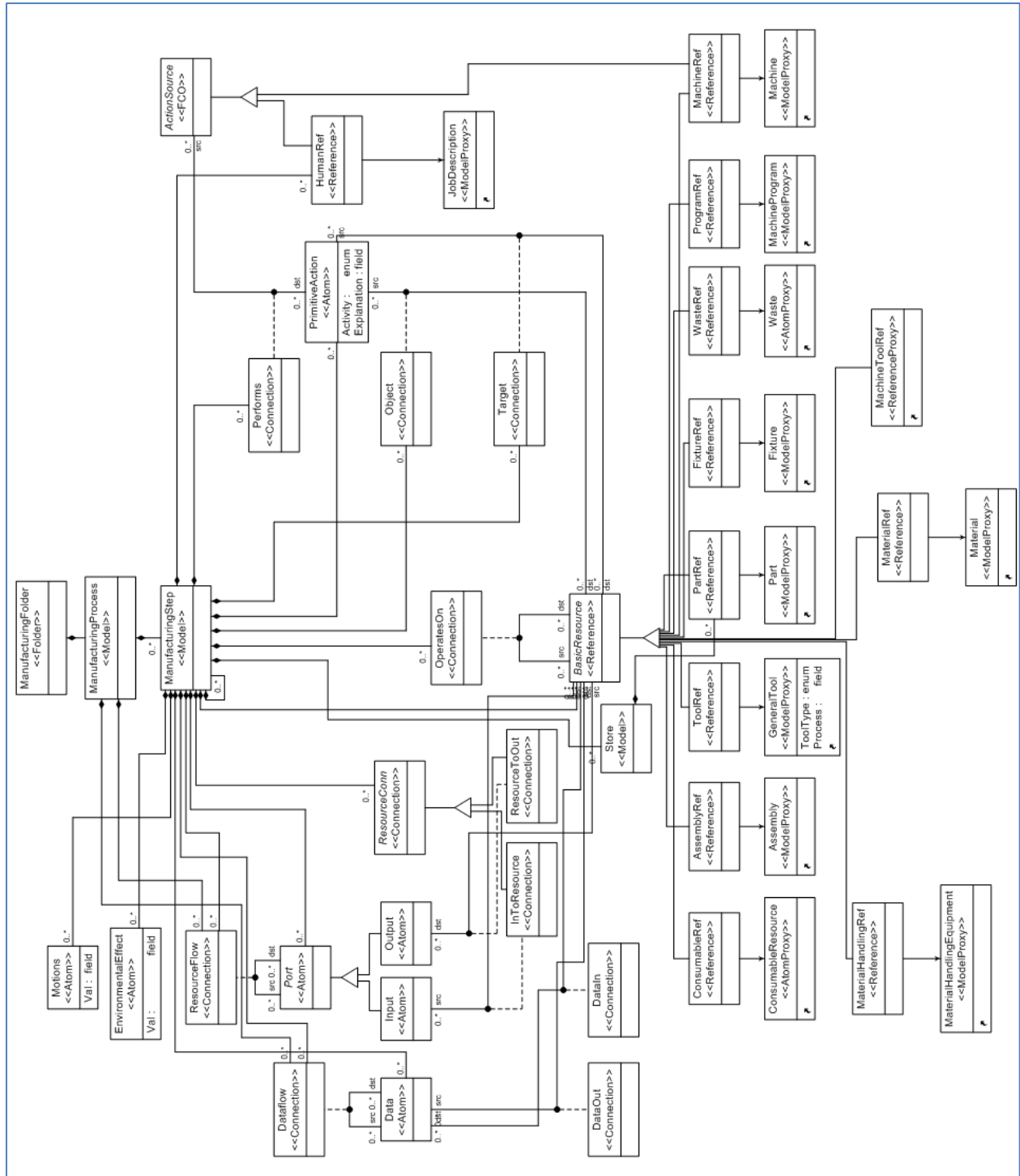
Figure 39. Metamodel for Manufacturing Processes

Each step has some key ingredients:

- a sequence of Primitive Actions that are performed in a step
- actor(s) that perform the step and that can be human or machine

- target(s) that the step is performed on
- object(s) that is/are involved in a step
-

Targets and objects are generic resources: tools, machines, parts, assemblies, etc. The key elements of the model allow the transcription of natural language sentences into a graphical form, where icons (nodes) represent entities and edges represent relationships.

3.4.2 Models Created Using MCPML

Using MCPML language described above, one can create various models for representing the foundry, the human assembly processes, as well as manufacturing processes. In this sense, a model is a concrete description of a specific manufacturing capability, conforming to one of the metamodels described above.

The models for machines, tools, machine tooling, consumables, fixtures, human resources and skills are used for representing what is available in the foundry. These models are representations of the data elements currently present in existing data sources such as company databases and machine data sheets, among others. Obviously, tools can be built that automatically populate (w.r.t. these aspects) an MCPML model with the content of such company databases. The key point here is that in MCPML one can express models that describe the capabilities that the foundry can provide. Capturing this information is important because it allows one to decide whether a design can be manufactured in a given foundry.

Manufacturing process models consists of steps, where primitive steps have the components: action(s), actor(s), target(s), and object(s). The figure below shows an example: a step in a drilling process where the worker (actor) moves (primitive action) the table (object). Such graphical programming can be quite powerful, easy to read, while providing a formally (i.e. computationally) analyzable representation of manufacturing processes. Several examples have been built: drilling and milling processes were modeled. As described above, manufacturing process steps are hierarchical, i.e. can contain sub-steps of arbitrary depth, and they can be arranged in a sequence. In other words, they can be refined into an arbitrary level of detail; e.g. the drilling step had 60+ sub-steps.
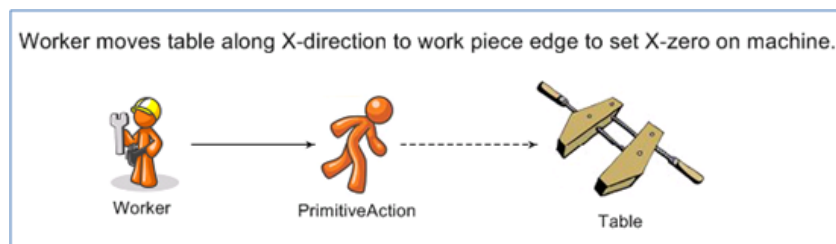
Figure 40. Manufacturing Process Step Example

Human assembly process models focus on human activities. They are similar to manufacturing process models, but they capture human postures and activities on a very fine level of detail. We have populated a suite MCPML models with a collection of generic models for human assembly

Approved for public release; distribution unlimited.

tasks and activities. Such models can serve as a 'dictionary' that industrial engineers pick from when creating instances of the human assembly steps for manufacturing a particular design.

Though the design is not part of the manufacturing library, manufacturing models frequently reference design elements as part of their definition (e.g., as the object of process actions). Therefore, the MCPML does allow references to generic design models elements. The metamodel for CyPhyML META designs, developed by the Vanderbilt META team, is included for illustration.
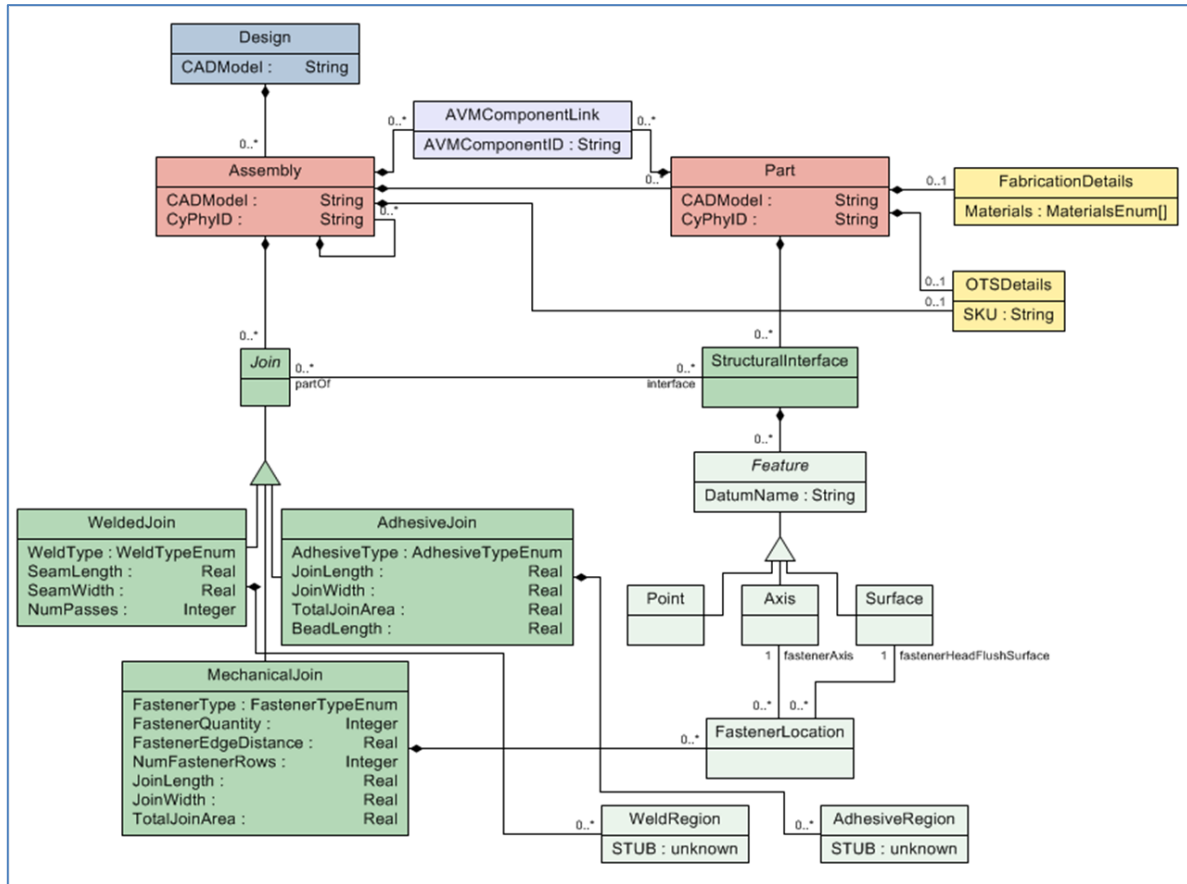
Figure 41. Metamodel for META Design models

Approved for public release; distribution unlimited.

3.4.3 Use Case for MCPML

MCPML is implemented as a visual modeling language (supported by the GME tool), but it is more general than that. In fact, MCPML models can be stored in a database, and database query languages and analysis tools can be used to perform various analyses.

The envisioned main use cases of MCPML are as follows.

Library Population:

1. A foundry industrial engineer populates the MCPML with manufacturing process models and human assembly models. Both contain steps, where each step contains actions, actors, objects, targets, possibly sub-steps, as discussed above. Human assembly processes contain instances of the specific steps from the human assembly process model library.

Foundry Configuration:

1. The designer creates a design and makes the result of the process available as META Design models, loaded into an MCPML model database.
2. The industrial engineer analyzes the design and runs various queries against the MCPML library and its capability models to determine if the foundry has the right capabilities for manufacturing the design, and to define the required foundry configuration for manufacturing the design. This could be an iterative process that is partly concurrent with other activities below, and should be highly automated (using tools from iFAB TA 1 and TA 3).
3. The industrial engineer uses various analytical and simulation-based tools to verify the process models. Tools check the soundness and completeness of the process, verify that resources are available and capable, simulate the entire process with potential variances and estimate production rates, etc.
4. Finally, the industrial engineer generates assembly instructions and a detailed manufacturing process plan from the models – for execution on the foundry.

Note that the above use case just captures the main activities for MCPML: library population and foundry configuration. There are also other library use cases:

Manufacturability Analysis:

1. A designer uses iFAB manufacturability analysis tools that use MCPML capability models to analyze his design to determine if it needs to be revised to achieve manufacturing goals.

Foundry Monitoring and Adaptation:

Approved for public release; distribution unlimited.

1. During operation, based on experience, it may be necessary to modify the manufacturing process. These modifications are to be done on the manufacturing and human assembly process models in MCPML, followed by verification and then execution.
2. The MCPML models can also be used to configure other manufacturing information systems (e.g. production flow monitoring systems). The advantages of using a model-based approach to do this have been demonstrated in the past [25].

3.4.4   Tools

MCPML is a domain-specific modeling language supported by a modeling environment (GME). The modeling environment has been configured and used for building prototype models. Several other tools have been designed and being developed:

- Tool for importing the META Design models (that are compliant with the metamodel for MCPML Designs)
- Tool for importing spreadsheets that capture the human assembly process library
- Tools for populating a SQL database from MCPML models
- Prototype queries that illustrate what sort of (simple) analysis questions can the MCPML models can answer
- This is just an initial set of tools, but this set can be expanded to provide better integration of MCPML with other systems.

3.4.5   Summary

MCPML is a domain-specific modeling language for capturing the manufacturing capabilities of a foundry where products described using META TDPs (and most likely designed according to a META design process using META design tools) are manufactured. The language is used to build extensive models for foundry capabilities and libraries of human assembly processes to enable foundry operators to rapidly configure the foundry for a particular design. The concept of rapidly constructing these manufacturing models and linking them to a design model, while also providing a platform for analyzing the models before physical execution in the foundry, is very powerful – it can potentially speed up the path from design to completed construction.

3.5 Define Virtual Manufacturing Environment

As part of our iFAB effort, the Boeing team developed a Virtual Manufacturing Environment that provides a multi-user capable avatar based virtual world for prototyping a factory layout and manufacturing and assembly processes to be performed in the foundry. The VME allows users to interact with the foundry, its facilities, and the products being manufactured. The user can also modify the layout to prototype virtual foundry configurations by directly interacting with the virtual environment and saving it for later use. The environment supports avatar bots (computer controlled avatars) via scripting to perform foundry operations.

3.5.1 VME Requirements

As part of developing and defining the VME we performed a study to identify what potential VME users would like to see in a VME beyond the initial concepts above. The approach taken was to conduct a series of interview of people representative of potential future users. This included engineers at GM and Boeing, as well as graduate students at two major Universities who were working on virtual simulation projects. The interviewees would be exposed to the VME concept in live interviews and their thoughts and reactions would be discussed and probed. The results of the interviews would then be reviewed to identify common themes and requirements.

3.5.1.1 Approach

The GM and Boeing interviews were structured to address the following questions:

- What are the current issues and challenges to updating the assembly process?
- Do the players see value in a VME type system?
- If so, what is the value they see?
- What do they want a VME to do?
- What features would users be looking for?
- What would have to be addressed to achieve success in:
    - Implementation
    - Usage
    - Acceptance
    - Integration.

Interviews for this project were conducted with three distinct groups during the fourth quarter of 2011 and January of 2012. The first group consisted of GM Employees and one researcher at TARDEC (one-on-one interviews with 12 engineers from various functional areas). The second group consisted of Boeing Employees (two focus groups of engineers on site at the St. Louis production facility). The final group included University Advanced Engineering Students and faculty (on the campuses of Carnegie Mellon University and the University of Michigan.).

The Boeing and GM interviews followed the same general outline. An initial segment introduced the topic. The topic description was deliberately vague so as not to bias the discussion. However, respondents were told that the current software that is available is useful for displaying or assessing defined process but not for use in creating and/or changing an assembly line. Respondents were then told that the purpose and objective of the discussion was to talk about

potential uses of VME software that was intended to simulate the design/change of manufacturing assembly processes. They were also told that that the discussion would focus in on identifying desired features and benefits of a more dynamic, flexible, collaborative VME of the future.

The phase of the interview sought to elicit "wish lists" based on current experiences with line setup and line changes. Before talking about the VME software directly, respondents were asked to describe their experiences with production line setup/changes with regard to issues and challenges. Questions were used to seed discussions. Examples include "Give me a recent example of a line setup or a major changeover you were involved in?" and "What issues did you encounter and how could the process (not the line itself) been made better?"

The final phase of the interviews focused on eliciting potential desirable features of a VME. The description was detailed enough to communicate the general idea, but not much detail was provided so the respondents would not get side-tracked on specifics. The key takeaway from the description was that respondents should think of it as a 'blank slate' tool for development. Two current simulation programs were references as "sort of" examples, 2nd Life and Sim City.

3.5.1.2 Results

Overall, interest in the VME idea was very high. Among both the 12 GM employees and the Boeing engineers we interviewed for this study, a comprehensive virtual simulation of the product design and general assembly process was seen as a very valuable tool at both companies. Many of the reasons for interest overlapped, while others were specific to the individual company.

Key drivers of interest that engineers at GM identified in the VME idea were:

- Improves communication across plants and disciplines
- Improves access to information for all involved parties
- Provides system-wide compatibility
- Shortening of development and assembly times by identifying problems earlier
- Saves time and money by substituting virtual simulation for physical mock ups and site visits
- Allows virtual testing of difficult ergonomic issues

Key drivers of interest that engineers at Boeing identified in the VME idea in addition to these were:

- Shorter cycle times by facilitating adoption of their move to the "Pulse" assembly process
- Big step toward their goal of increasing the level of standardization in the production process
- Better baselining and capture of information

While each of the engineers we interviewed had their own reasons for liking the VME idea, we were able to identify three basic/overall system capabilities that drove much of the value all of

the engineers saw in the system: The Value of Information, The Value of Communication, and The Value of Reducing Trial & Error Loop-Back Iterations.

As the interviews in this study illustrate, and it certainly comes as no surprise to anyone at GM or Boeing, it takes a massive amount of data and knowledge to assemble a vehicle or an airplane. We collectively refer to this data and knowledge as just "information". The starting point for any assembly design, be it a new install or a modification, is gathering a large amount of information.

Due to the size and nature of the task, getting this information in an optimal way is often no easy matter. To obtain it, the engineers on the planning team have to navigate a gauntlet of variables that affect the outcome. He or she needs to know what info they need, where it's at and how to get it. It has to be gotten in a timely way, it has to be accurate, and if it's info that changes, it has to be current. Ideally, it should also be in a format that's easy to access without having to translate it or re-enter it. In fact, one of the primary sources of value the engineers saw in this tool was that they assumed VME could solve many of the "information problems".

Knowledge and data are obviously the foundation of an efficient and effective assembly design/ modification process. However, the existence of even perfect information is useless unless it's communicated in a timely and effective manner.

The engineers we interviewed said that poor communication is one of the root causes of assembly design problems that arise from the lack of good and proper information, as described in the section above. They saw the VME as being able to effectively address many current communication issues that result in the use of inaccurate or incomplete data being used for decision making. The significant improvements in communication that a VME system would offer was the key method for improving data issues and another major source of value the engineers saw in this idea.

The engineers in this study confirmed that with product and assembly design & development, they always strive to make it a fast-moving, efficient, parallel-linear process. In theory, a purely parallel-linear product assembly development process (e.g. overall vehicle design development, individual part development, tool development, assembly/manufacturing process development, ergonomics and assembly plant decision) would be the fastest path.

The reality of development is different. Because of the inter-connectedness of the various processes, product & assembly development is actually a 'rolling and looping' process where developments/changes in one area impact decisions in many other areas. The complexity and inter-connectedness of the decisions means that many different groups need to review & have input on the process at the same time, which often causes delays. This necessitates an iterative process of countless 'course corrections' as engineers must 'loop back' to make changes which then have a cascade effect in other areas.

Accurate simulations of the type suggested by the VME system would be an effective substitute for trial & error. Mistakes revealed in the virtual world are adjusted for quickly and at little cost. Problems are identified before they are built. Solutions can be tried and tested without getting caught up in the costly and time consuming parallel loop-back process. The ability to reduce trial & error loop-back iterations is the third major area of value the engineers in our study saw in the

Approved for public release; distribution unlimited.

VME simulation idea. Note that the AVM approach to this issue during design (and the overall product lifecycle) is to break with the cycle via correct by construction, rather than trying to make the iterations less expensive. On the other hand, virtual iterations can be viewed as part of the correct by construction process since they do avoid "physical" iterations.

When defining their idealized view of the VME, engineers from both GM and Boeing had no trouble identifying numerous specific features or capabilities they would like this virtual simulation to have. They were also able to describe the benefits that each feature would offer. In most cases, the respondent would relate the feature and benefit to an issue or issues with the current process they had described earlier in the interview.

After documenting and analyzing all the features and capabilities respondents brought up, we grouped them into one of five functional categories that were determined based on how they logically related to each other. Some features were important in multiple functional categories and are listed each time. Features that didn't seem to belong into one of the five categories were put into a sixth "other" group at the end. Thus the six groupings of features are as follows:

1) **System Qualities**: Compatibility / Versatility / Adaptability / Usability
2) **Simulating Reality**: Substituting VME Simulation for Trial-&-Error Loops
3) **Communication**: Interfacing / Sharing Knowledge / Collaboration
4) **Baselining**: Capturing & Storing Current Data & Information / Building On Experience
5) **Ergonomic Analysis**: Specialized Subset of Assembly Issues and Requirements
6) **Other Benefits and Features** of a Serious Game Simulation

In addition to the benefits and desirable features, the interview also captured potential issues and pitfalls associated with using a VME. These were summarized in the Quality Function Deployment diagram of Figure 42.

Approved for public release; distribution unlimited.

Serious Games Simulation QFD Grid

The figure below is a Quality Function Deployment (QFD) matrix. The vertical axis (rows, grouped by category) lists the design requirements / system features; the horizontal axis (columns) lists the customer issues/needs. A "+" marks a relationship between a feature and a need.

Design requirements (row labels, by category):

- **System Qualities:** All Inclusive / Primary System; Central Database or Logical Access; Data Security /Integrity Built In; Plant Flexible / Proxy Plant Options; Product Flexible; Easy to Learn / Use (Intuitive); Quick/Responsive; One Common Software; Modular (e.g. MS Office)
- **Simulating Reality:** Hi Resolution 3D Visualization; Can Simulate Whole Line; Simulate Line Dynamics
- **Communication:** Advanced Interactive Capabilities; Collaboration Tools; Provide for Vendor Participation; Operates Globally; System Wide Compatibility; Bridge "Parallel" Operations
- **Baselining:** Primary Mechanism for Baselining; Capture Usage; Automatic Data Saving; Data "Gatekeeper"; Include "Baked In" Manufacturing Rules; Parameter Library (Tools & Parts); Benchmark / Metric Library; Track Performance / Benchmarks & Metrics; Training/ Learning Mode
- **Ergonomic Issues:** Handle Ergonomic Issues; Avatar Capable
- **Other:** Advanced Reporting Capability

Customer issues (column labels):

- Information is Decentralized / Hard to Get
- Information: Not Accurate / Timely
- Incompatible Systems / Software
- Data Security Issues
- Access to Info / Tools is Uneven
- Many Different Software Used
- Ambiguous Software Results
- People Elect Not to Use Some Software
- Issues Too Demanding for Software
- Software Technical / Hard to Learn/Not Used
- Limited 3D Software
- Too Many Loop-Back Iterations
- Problems ID Too Late in Process
- Over Reliance on 2D / Math Sims
- Need Physical Mock Up To Fix Issue
- Plant Decisions Made Late
- Long Cycle/Pulse Time (Boeing)
- Communication: "Out of Phase"
- Complexity Makes Improvements Difficult
- Hard to Shorten Dev. Time/Into's Problems
- Many Unique / Non-Standard Tools
- Making All The "Parts" Work Together
- Part Designs Not Optimized for Assembly
- Fit / Tolerance Hard to Address
- Need for Mock Ups / Deal With Motion
- Large Tools / Assembly
- Communication: Between Locations
- Communication: Internal (Silos)
- Difficult for Vendors to be Part of System
- Difficult to Get Operator Input
- Baselining Inadequate/Lack of Current Docs
- Tribal Knowledge Lost
- "Experience" Learning Hard to Capture
- Documenting is Difficult / Inconsistent
- Long Learning Curve for Assembly Design
- Ergonomic Issues Difficult/Time Consuming

Figure 42. VME Requirements Quality Function Deployment Grid

3.5.2   VME Implementation Approaches

Using a 3D serious game framework based on a multiplayer virtual environment for the VME provides a basis for a number of collaborative activities and supports the objectives identified above. The multi-user virtual environment supports distributed planning and analysis of manufacturing and assembly workflows in a very visually rich manner in real-time. It supports realistic planning and analysis of collaborative workflows by permitting simultaneous access by all of members of a multi-person work cell regardless of their geographical location. Avatar Bots (computer controlled avatars) may also be created to work alongside of or in place of human controlled avatars. Given the rapid rate of technological advances in the underlying gaming (and entertainment) area, using serious games as a basis of the framework provides early access to solutions to some of the technology challenges encountered in existing conventional approaches – for example, realistic computer controlled human posturing is clearly a desirable feature for gaming and other entertainment oriented 3D models, and solutions in that domain could be exploited for iFAB and MCPML.

Once such a framework was built for iFAB manufacturing capabilities, the same framework could be used for a number of purposes such as virtual training on new configurations. The virtual environment would provide a realistic, immersive, highly interactive synthetic manufacturing practice experience to the foundry workforce to enable them to more rapidly progress along the learning curve. The result would be an experienced workforce before manufacturing even begins.

Consider two families of 3D worlds that could serve as architectures for the serious games facet of an MCPML model. The first of these, known as a virtual world, as typified by Second Life, is focused on providing a large and massively shared virtual world with content that can be developed and contributed by any user. The second is a video game engine, such as those used in the world of first person shooters. Each has certain advantages when viewed from the perspective of modeling human aspects of manufacturing capabilities.

Second Life is the most well-known 3D virtual world with millions of simultaneous users from all over the world collaborating together. It was developed by Linden Lab and went live in 2003. Second Life has proven sufficiently popular that in-world artifacts and activities have been used to generate real-world economic benefits and many organizations have established presences in Second Life for a variety of purposes, including marketing, collaboration, and education. Several countries have even established "embassies" in Second Life as a means of providing online access to some embassy services. The US Federal Government has several departments that regularly conduct business meetings, the US Air Force conducts virtual tours of real facilities in Second Life, and the National Oceanic and Atmospheric Administration (NOAA) federal agency has very interesting interactive models of oceans and weather.

More importantly from an iFAB perspective, supporting the sort of collaboration activities indicated above as primary and secondary benefits of a serious games MCPML facet, a virtual world like Second Life provides mechanisms for defining both visual and behavioral aspects of modeling spaces, objects, and humans. Such a mechanism could be used to define the appropriate manufacturing capability attributes in the MCPML language.

Second Life does have a number of limitations that would have to be addressed in any iFAB serious game implementation. First among these is access management. Second Life is publicly and freely accessed by anyone anywhere in the world that has the free client installed. Though it is anticipated, given the inclusion goals of AVM, that there be low barriers to entry and wide access to MCPML (and other iFAB capabilities), having some control is desirable. Because Second Life is owned and operated by a private company there are many other concerns. There are alternative environments to Second Life, such as ProtonMedia's ProtoSphere, which is a Commercial Off-the-Shelf (COTS) product that can be used within a managed context. ProtoSphere is typically intended for business usage in contrast to the more entertainment or education market oriented Second Life approach. Second, though they are extensible, the default interaction modes provided by Second Life (and similar environments) are geared more towards social interactions between humans mediated by their avatars rather than accurate modeling of the more sophisticated and complex activities characterizing manufacturing capabilities and processes.

The second family of environments, those based on game engines used first person shooters and similar games, does not suffer that second limitation. Such game engines offer ever increasingly "realistic," expansive, and inclusive game worlds. Though there are simplifications and approximations, the desire to create the most realistic games drives game engine developers to rapidly adapt new technologies and push the envelope so that they have the potential to provide a distributed virtual environment at a level of abstraction sufficient for the most demanding iFAB purposes. While most of these games engines are very expensive, t. There are a number of open source game engines providing a diversity of candidates that could serve as the basis for the MCPML serious games framework. These are often open source releases of previous generations of successful proprietary engines, with the various "Doom", "Quake" and "Unreal" engines being typical. Many of these support import and export of standard 3D file formats such as COLLADA or X3D, which facilitates integration with the engineering design tools such as CATIA that may be used to create the original 3D design and manufacturing models. Although these game engines provide higher levels of fidelity than virtual worlds, these engines are made to prohibit cheating by game players. These anti-cheat mechanisms severely restrict interfacing with other applications and external scripting, two requirements of the iFAB framework. Additionally, these game engines achieve greater performance with higher fidelity by installing all the assets on every client computer. On the other hand, a virtual world centralizes all the assets on the server, so when a new asset is added or an existing model in the virtual world is modified, everyone connected immediately sees the change. This capability found in virtual worlds, not game engines, is a highly desired capability for collaborative design. Another issue of note is that game engines, even in advanced games, generally do not support specialized hand postures associated with using particular tools. Cut scenes, which are pre-made videos (not generated in real-time), are a typical way to display such details. In order to support hand postures in a generic way, so that the hands can be used with any tool or part, it is not practical to use the technique of cut scenes that are videos.

Proprietary commercial virtual environments capable of using CAD models are also starting to be offered through companies like Dassault. However, such products are frequently very expensive, and usually encourage the use of proprietary data formats instead of the open formats typically found in rapidly evolving open source tools.

Approved for public release; distribution unlimited.

3.5.3 Modeling Manufacturing Capabilities

We explored the use of various serious game approaches to defining an MCPML framework defining 3D virtual reality models for manufacturing capabilities and processes. These models will be used to support planning and analysis of foundry configurations, in particular to support the planning and analysis of human activities within manufacturing and assembly process. We believe such a representation will provide capabilities required to ensure that the resulting foundry configuration is realizable in the sense of being able to sustain production of the target design. Specifically, by enabling the development of a "realistic" and accessible virtual environment depicting the configured foundry, or foundry subset, a number of different analysis modalities will be enabled that can provide varying levels of confidence in the result of the configuration process.

These analysis modalities include "3D walkthroughs" by foundry worker proxies to assess the viability of the proposed configuration from both producibility and ergonomic perspectives. That is, does the proposed configuration enable the foundry worker to perform the proposed task, and can the task be performed with the projected rates while satisfying ergonomic requirements. The first could involve either inspection, or an actual virtual execution of the manufacturing process. The second interprets the first in the context of the proposed schedule. The foundry worker proxies could be the actual foundry workers themselves, representative foundry workers (e. g. team leads), supervisors, or industrial engineers or other members of the manufacturing engineering staff, or by members of the foundry configuration team. The analysis can also be performed in a more automated fashion, e.g. by improved computer controlled avatar bots.

4.0   RESULTS AND DISCUSSION

This section discusses the results obtained from applying the schemas developed under the characterization tasks to the example assembly problems.

4.1   Human Assembly Process Capability Examples

A number of assembly operations were analyzed for human process capability using the human assembly process model. The operations analyzed include the iFAB Meta example and the GM engine marriage, instrument panel installation and engine dress.

Each element in the Human Assembly Process Model provides detailed information about a manufacturing task, given that some high level information (Activity, Object, Plant Equipment, Material or Tool) is known about the manufacturing process. To employ the model, information about the assembly operations is used to select the specific model entry by specifying information for the various attributes of the model.

The next subsections provide detailed walkthroughs of the application of the Human Assembly Process Model developed under this project. This manual walk-through using a spreadsheet representation of the model (Appendix A) is a necessary precursor to the automated assembly service provided in the MCPML interface tools and serves as a storyboard pseudo code to implement the heuristic assembly process in a manufacturing API. Through that MCPML API, other iFAB performer tools can query the library and ascertain useful deductions on the underlying manufacturing model library.

The first example provides a detailed overview of how to use the model using the iFAB META example and based on the technical data package (TDP) supplied by Vanderbilt University and DARPA. This is followed by a set of more compact examples based on GM processes. The GM examples use the product and tooling data given in Tables 1and 2.

4.1.1   iFAB Meta Example
This example using the Meta design provided by DARPA (Figure 43) is a manual walk-through of the human assembly process model applied to the design subassemblies.

Approved for public release; distribution unlimited.
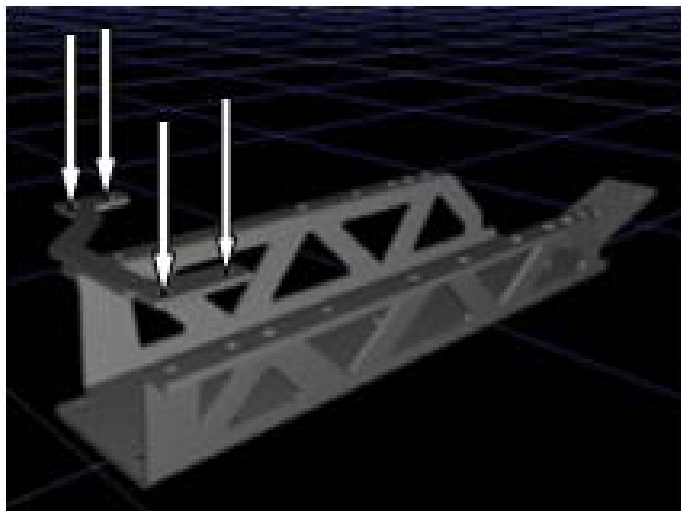
Figure 43. DARPA iFAB Meta Example

Assume we are given some high level process information like "Assemble" Rear Brace to Main Chassis. The high level process information will be used as an input for the selection of Activity (first column) in the taxonomy shown in the Figure 44. Start with the Activity column which is the first column of the "Taxonomy" tab.   Filter the column to show the tasks for "Assemble".
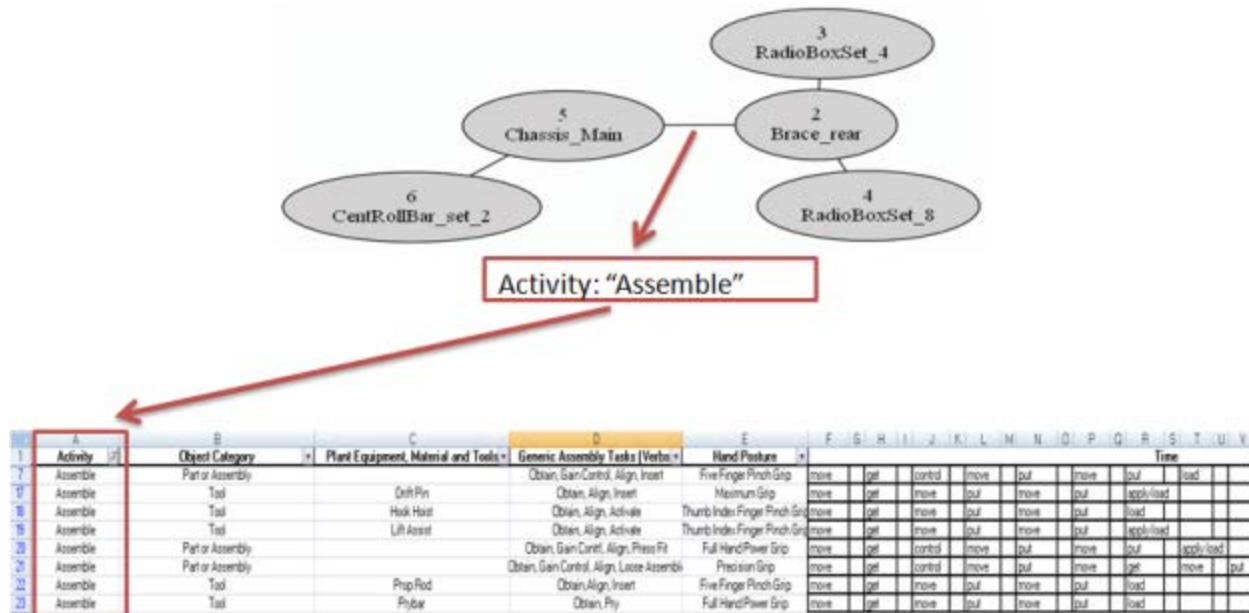
Figure 44. META Example to Assemble the Rear Brace to the Main Chassis

Next in column B select a Part or Assembly or a Tool if one is needed for the assembly task. Continuing the META case, select "Part or Assembly" to assemble the rear brace. This narrows the search further now displaying only the following tasks (Figure 45).

Approved for public release; distribution unlimited.

Figure 45. META Example to Specify Assembly of Part

At this point, the lower level General Assembly Task (column D) detail in the taxonomy is used to describe the process. This requires some choices to be made by the user and must be reduced further using additional information related to the process requirements and the station layout in the virtual environment. From the remaining three Generic Assembly Tasks listed, the one that is most relevant in the META example is "Obtain, Gain Control, Align, Insert". This was chosen because we first obtain the rear brace, gain control of the brace, align it to the chassis and insert it onto the frame. Note that in this case the word "insert" used to denote the final micro elements (move, put, load) that will be used for the MODAPTS calculation.

Figure 46. META Example to Specify Generic Assembly Tasks

Copy the task row to the Process Tab, and then complete the following assessments.

4.1.1.1 Time Assessment

The micro elements for each verb in the process are listed in columns F through AL. They were pulled from the "Generic Assembly Tasks" tab. The verbs are listed in column A and the related micro elements follow in the respective columns. For example, the micro elements for "Obtain" are "move, get". The micro elements for "Gain Control" are "Control" and the micro elements for "Align" are "move, put".

Approved for public release; distribution unlimited.
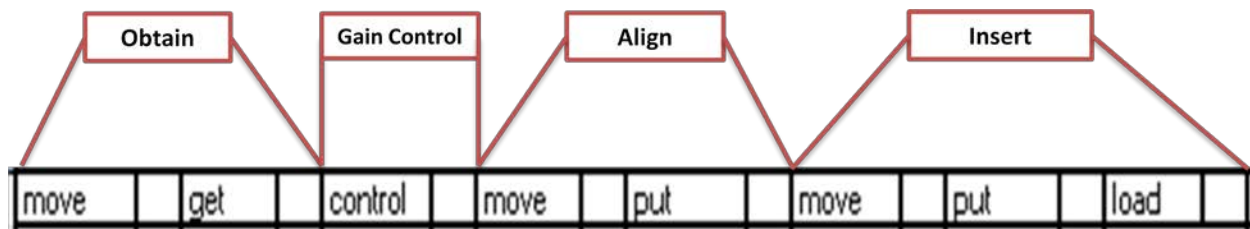
Figure 47. META Example Specifying MODAPTS Breakdown

The next step in determining the MODAPTS calculations is filling in the spaces following the micro elements in columns F, H, J, L, N, etc. In order to obtain these values, refer to the "Task Metrics" tab. A decision based on additional information related to the process requirements and the station layout in the virtual environment is needed to determine which description best completes/represents the manufacturing task.

For instance, what type of "move" and what type of "get" would best represent the micro elements involved for the "obtain" portion of this task. The best choice will depend on how far away the brace is per the layout. Assume the brace is within arm's reach of the operator. We look in the "Task Metrics" tab and in column A we find "Move" and in column B we see the different types of moves and the times associated with them. Since we assumed that the brace is within arm's reach we select "move arm" and we have the associated time that it takes to move the arm (0.00645 minutes). The next step is to decide what kind of "get" should be chosen. Again, in column A of the "Task Metrics" tab we look up "Get" and see that there are three different "gets" that can be chosen. Let's assume that in order to obtain the brace it requires a "Get Two Fingers" and the associated time with that is 0.00215 minutes.

If we continue using this process for "Gain Control", "Align" and "Insert" we get something that looks like the figure below.

Figure 48. META Example to Populate MODAPTS Break Down

The total time is the sum of columns G, I, K, M, O, Q, S, U or 0.01935 minutes. This time can be compared to other alternative processes to determine the optimum time. The total time is 0.01935 minutes.

4.1.1.2 Hand Clearance, Reach and Strength Assessment

The most appropriate (or closest) hand posture for the task from the library of hand postures is given in column E. Pull the hand posture from the library and position it appropriately for the task & check for clearance (i.e. interference with surrounding objects mean check fails otherwise it passes).

Approved for public release; distribution unlimited.

Select a posture from the library based on the station layout and the position necessary for the worker to be in the clearance zone for the operation with hands lined up appropriately for the task. Note that at early stages in foundry configuration this information may not be available so that the model user may need to make estimates/judgments about likely values. The MCPML services that automate using the models contain built in heuristics based on engineering judgments for this purpose. Check for reach anytime an "obtain" or an "align" is performed (object within the reach zone passes, otherwise fail).

Compare required installation force (from META) with max force for posture based on task (from the library).
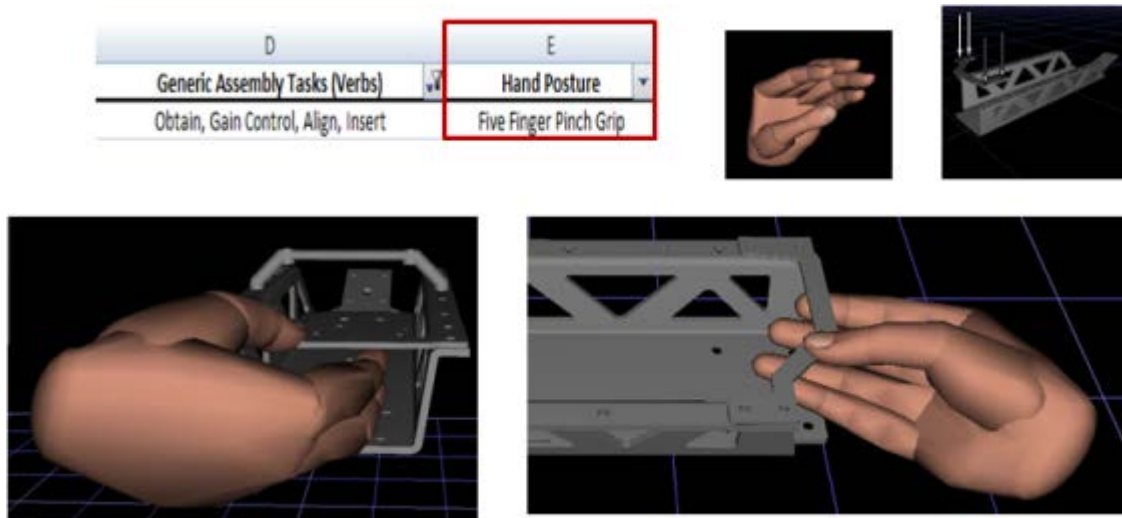
Figure 49. META Example: Specifying Hand Clearance
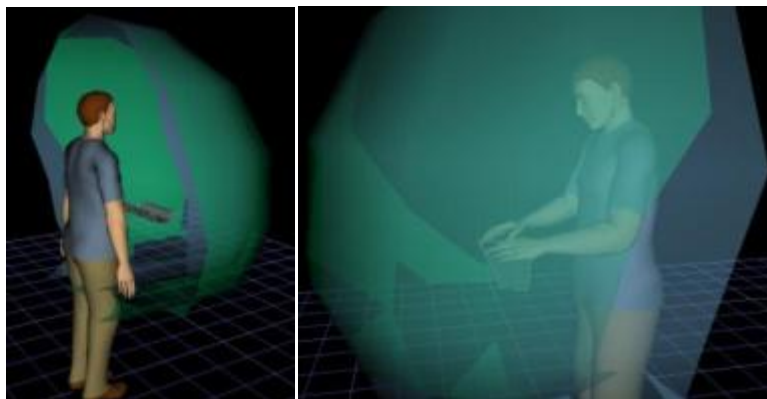
Figure 50. META Example: Assess Clearance Zone

4.1.1.3 Strength Assessment

Compare required installation force (from META) with maximum acceptable force for posture based on task. In this case, the maximum acceptable force is 81.0N in the direction shown in Figure 51.
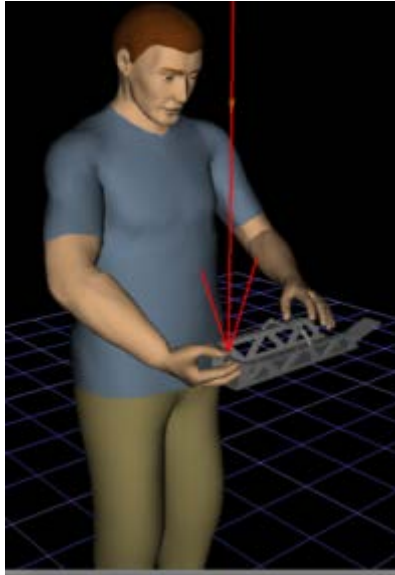
Figure 51. META Example: Assess Strength

4.1.1.4 Subsequent Assembly Steps

In the next step, one must fasten the radio boxes to the main chassis. Assume we are given some high level process information to "Fasten" Radio Boxes to Main Chassis (Following "Assemble" Radio Boxes to Main Chassis).

Approved for public release; distribution unlimited.
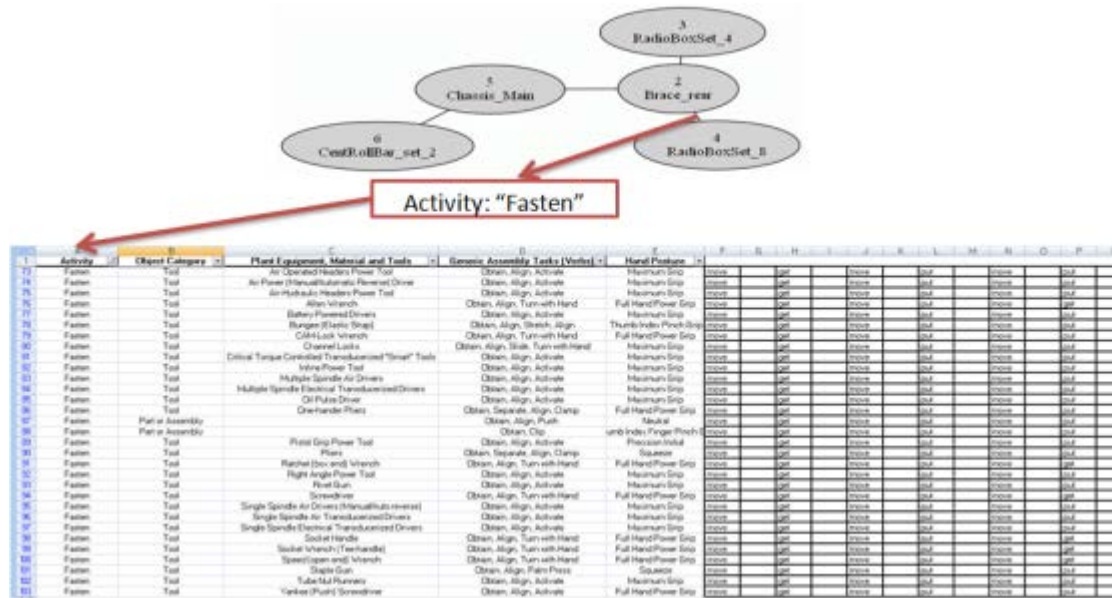
Figure 52. META Example: Specify the Fasten Activity for Radio Box to Main Chassis

Next, select the Object Category "Part of Assembly" or "Tool". A tool is required in this case, so select an appropriate tool.

Figure 53. META Example: Specify Using the Screwdriver for Fastening

The Generic Assembly Task associated with using a Screwdriver is: "Obtain, Align, Turn with Hand".
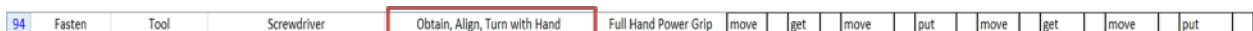
Figure 54. META Example: Specify the Generic Assembly Tasks

Approved for public release; distribution unlimited.

Using the "Task Metrics" tab and selecting the appropriate micro element times will provide an estimate of the time required to perform a fastening activity.
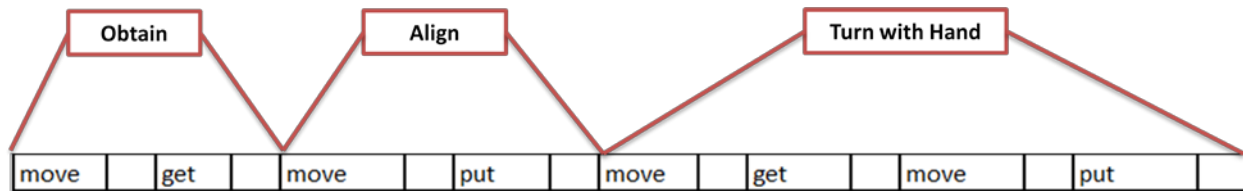
Figure 55. META Example: Specify the MODAPTS Break Down

The above process is then repeated for each assembly task and assessment capability required to complete assembly of the product. Though in principle the above steps can be performed manually, in practice, automation is required for integration with the automated iFAB foundry configuration process (as well as to scale to the foundry level). The MCPML provides services that implement the process described here that are suitable for automated foundry configuration.

4.1.2   GM Examples

In each of these examples, one or more operators perform one or more of the steps in a GM assembly task. In each case, the example describes the task, provides one or more figures to provide context, and then represents the use of the model to derive timing and other information in the form Activity → Activity Type → Tool/Product → Generic Task → Micro Task Sequence → Hand Posture, mirroring the process defined above of using the spreadsheet form of the human assembly process model.

4.1.2.1 Engine Marriage

The first operator on right side set will set the engine to the engine mounts by first obtaining the engine with lift assist tool. The first step will result in the following selections in the taxonomy.

- Install → Tool → Lift Assist → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Load → Thumb Index Finger Pinch Grip
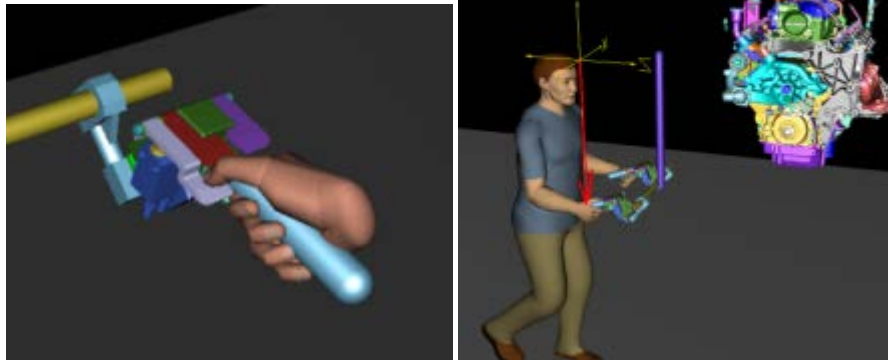
| 113 | Install | Tool | Lift Assist | Obtain, Align, Activate | Thumb Index Finger Pinch Grip | move | | get | | move | | put | | move | | put | | load | |

Figure 56. Obtain Engine with the Lift Assist Tool

Next the operator will travel to engine mounts with the engine secured by the lift assist tool. This results in the following selections in the taxonomy.

- Walk→ Human Motion→ Walk

The following selections are used for the engine marriage example:

- "Task Metrics": One Linear Foot = 0.00516 min.
- Walk ~15 ft
- 15 * 0.00516 = 0.0774 min.
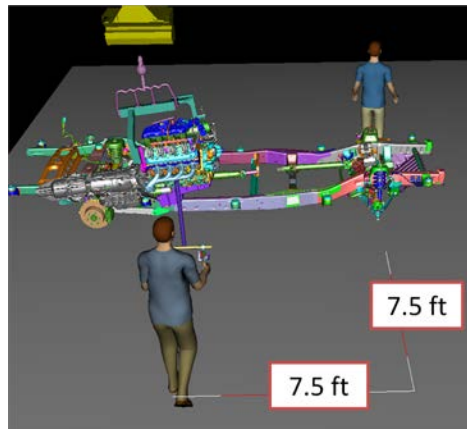
| 214 | Walk | Human Motion | | Travel | Neutral | walk | |

Figure 57. Walk with the Lift Assist Tool and Engine

Approved for public release; distribution unlimited.

Next the operator will the lower the engine and transmission assembly to the engine mounts. The following taxonomy selections will be used.

- Install → Tool → Lift Assist → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Load → Thumb Index Finger Pinch Grip
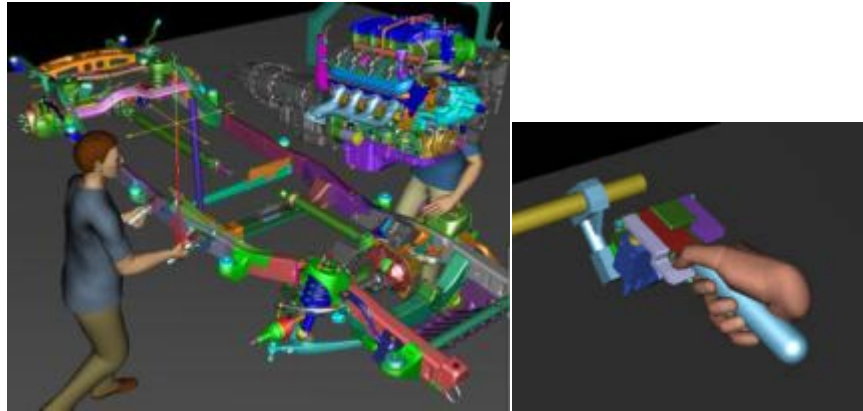
| 113 | Install | Tool | Lift Assist | Obtain, Align, Activate | Thumb Index Finger Pinch Grip | move | | get | | move | | put | | move | | put | | load | |

Figure 58. Lower Engine and Transmission Assembly Using the Lift Assist Tool

Next the first operator will secure the right engine mount. First a bolt is loaded resulting in the following selections in the taxonomy.

- Assemble → Part or Assembly →Obtain, Gain Control, Align, Insert → Move, Get, Control, Move, Put, Apply Load →Five Finger Pinch Grip

| 7 | Assemble | Part or Assembly | | Obtain, Gain Control, Align, Insert | Five Finger Pinch Grip | move | | get | | control | | move | | put | | move | | put | | load | |

Figure 59. Load Bolt to Power Tool for Engine Mount Secure

Then secure engine to engine mounts using the following taxonomy selection.

- Fasten → Right Angle Tool → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Apply Load →Maximum Grip
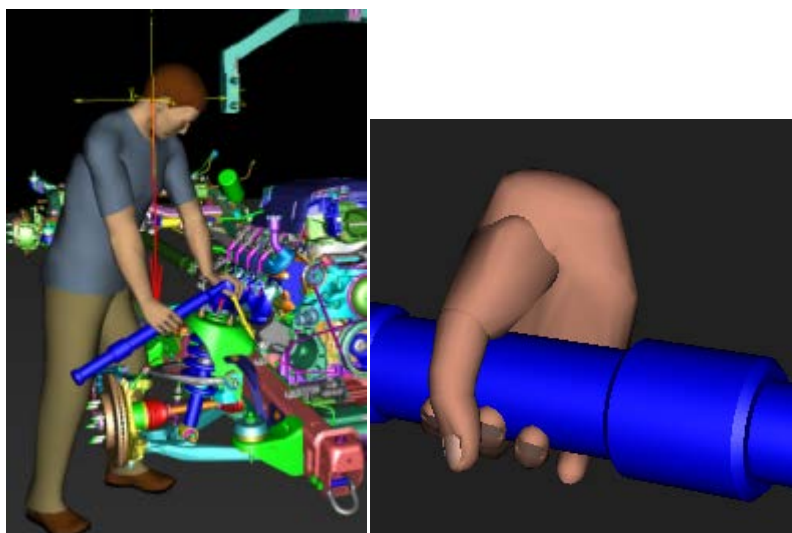
| 92 | Fasten | Tool | Right Angle Power Tool | Obtain, Align, Activate | Maximum Grip | move | | get | | move | | put | | move | | put | | load | |

Figure 60. Fasten Engine to Engine Mounts on the Right Hand Side

Approved for public release; distribution unlimited.

Then secure left engine mount by the second operator on the left hand side. First load the bolt using the following taxonomy selections.

- Assemble → Part or Assembly →Obtain, Gain Control, Align, Insert → Move, Get, Control, Move, Put, Apply Load →Five Finger Pinch Grip

7	Assemble	Part or Assembly		Obtain, Gain Control, Align, Insert	Five Finger Pinch Grip	move		get		control		move		put		move		put		load	
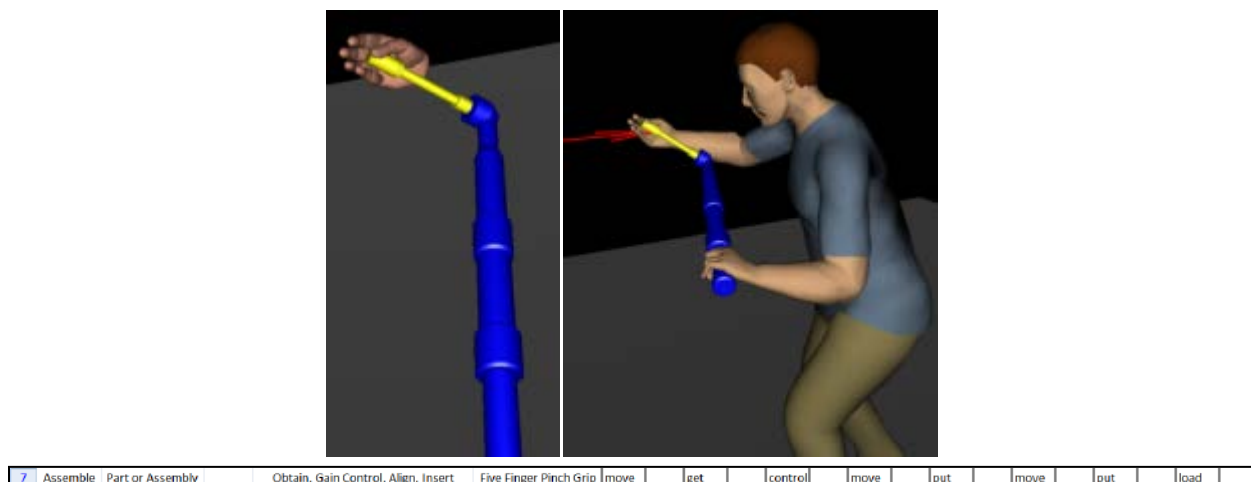
Figure 61. Load Bolt to Power Tool for Engine Mount Secure

And then secure engine to engine mounts using the following taxonomy selections.

- Fasten → Right Angle Tool → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Apply Load →Maximum Grip

92	Fasten	Tool	Right Angle Power Tool	Obtain, Align, Activate	Maximum Grip	move		get		move		put		move		put		load	
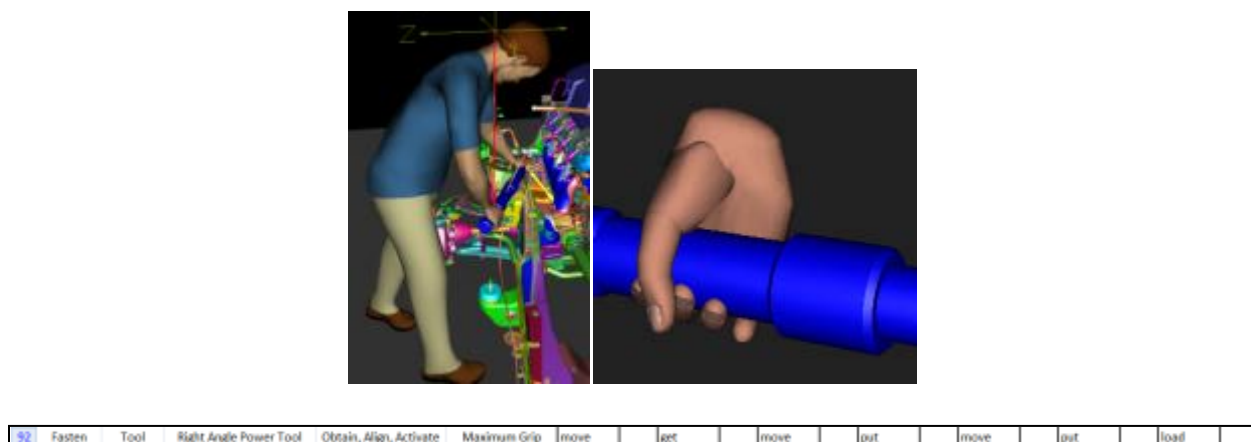
Figure 62. Fasten engine to engine mounts on the left hand side.

4.1.2.2 Instrument Panel (IP) Installation

The first operator on right side obtains IP from carrier. The obtain IP results in the following taxonomy selections:

Approved for public release; distribution unlimited.

- Install → Tool → Lift Assist → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Load → Thumb Index Finger Pinch Grip

113	Install	Tool	Lift Assist	Obtain, Align, Activate	Thumb Index Finger Pinch Grip	move	get	move	put	move	put	load

Figure 63. Obtain IP and Lift Assist Tool

Next, load IP to the vehicle cab using the following taxonomy selections. The following information is needed from the station mock up to complete the assessment.

- "Task Metrics": One Linear Foot = 0.00516 min.
- Walk ~9 ft
- 9 * 0.00516 = 0.04644 min.
-

Walk to the vehicle using the following taxonomy selections.

- Walk→ Human Motion→ Walk (Process Dependent)

214	Walk	Human Motion		Travel	Neutral	walk	

Figure 64. Walk to Vehicle with IP

Load the IP to the vehicle cab on the right hand side using the following taxonomy selections.

- Install → Tool → Lift Assist → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Load → Thumb Index Finger Pinch Grip
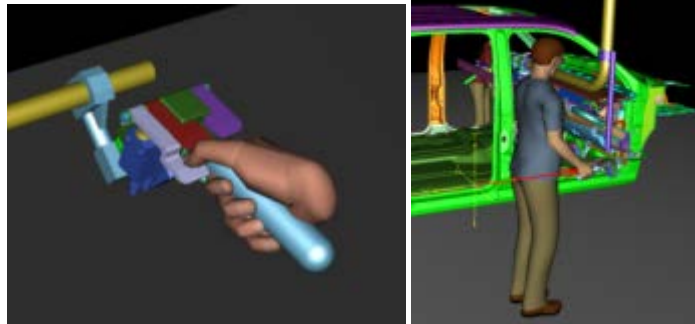
| 113 | Install | Tool | Lift Assist | Obtain, Align, Activate | Thumb Index Finger Pinch Grip | move | get | move | put | move | put | load | |

Figure 65. Load IP to Vehicle Cab on the Right Hand Side

At the same time, a second operator on left side assists with the IP installation and secure.

First, the operator assists with the IP load into the vehicle cab by helping to locate the IP. The following taxonomy selections are used.

- Assemble → Tool →Lift Assist → Obtain, Align, Activate →Move, Get, Move, Put, Move, Put, Apply Load → Thumb Index Finger Pinch Grip
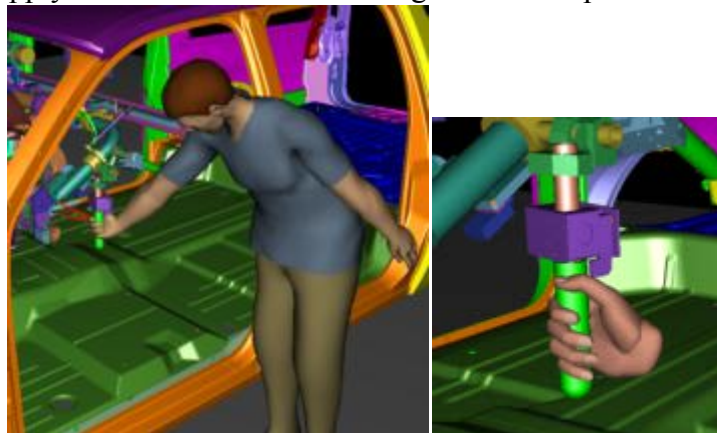
| 19 | Assemble | Tool | Lift Assist | Obtain, Align, Activate | Thumb Index Finger Pinch Grip | move | get | move | put | move | put | load | |

Figure 66. Locate IP on Left Hand Side

Then secure the IP to the vehicle cab on the right hand side.

- Fasten → Right Angle Tool → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Apply Load →Maximum Grip

Approved for public release; distribution unlimited.

| 92 | Fasten | Tool | Right Angle Power Tool | Obtain, Align, Activate | Maximum Grip | move | | get | | move | | put | | move | | put | | load | |
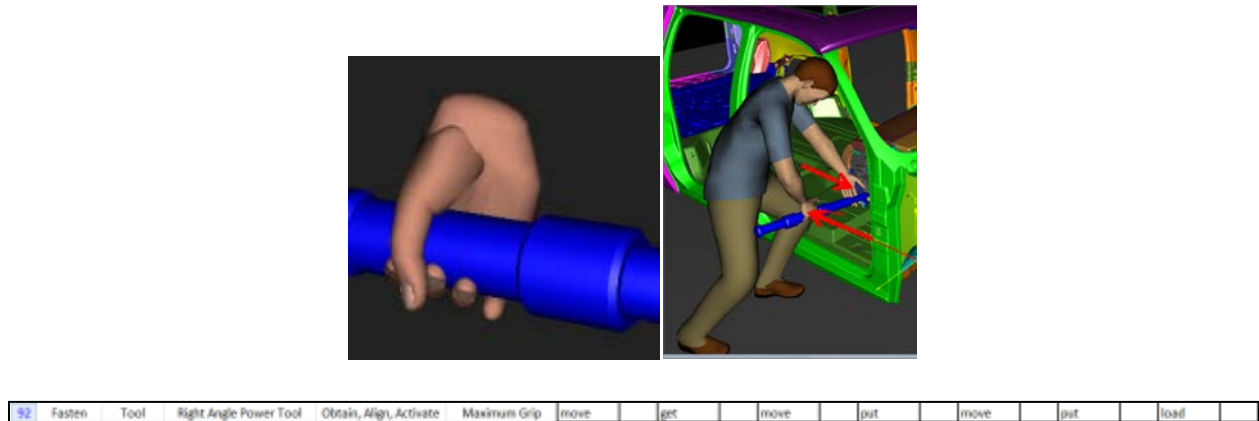
Figure 67. Fasten IP on the Right Hand Side

And the IP is also secured on the left hand side. First the operator loads a bolt using the following taxonomy selections.

- Assemble → Part or Assembly →Obtain, Gain Control, Align, Insert → Move, Get, Control, Move, Put, Apply Load →Five Finger Pinch Grip

| 7 | Assemble | Part or Assembly | | Obtain, Gain Control, Align, Insert | Five Finger Pinch Grip | move | | get | | control | | move | | put | | move | | put | | load | |
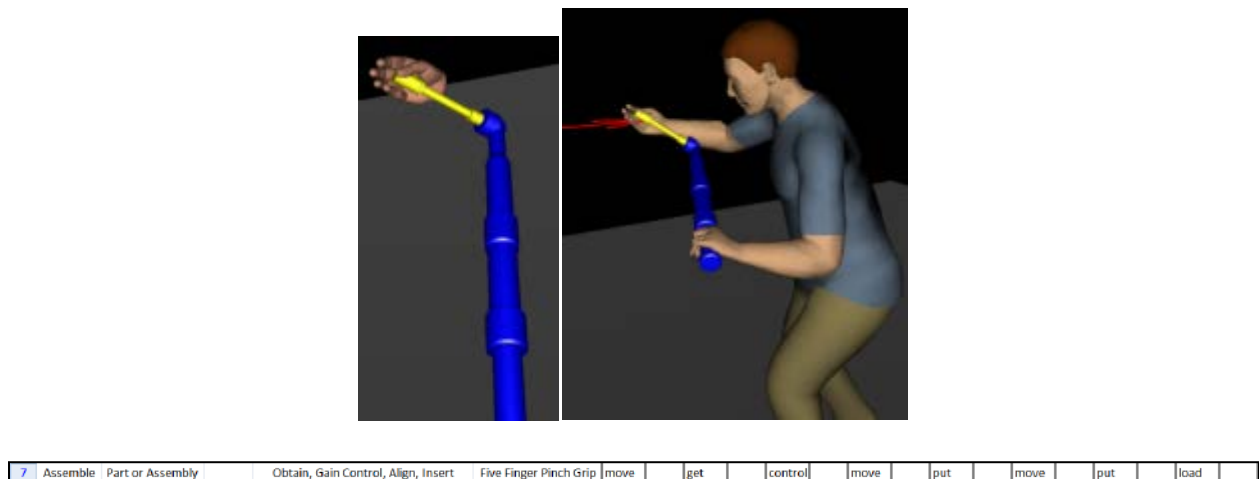
Figure 68. Load Bolts for IP Fasten

And finally the IP is secured to the vehicle cab on the left hand side using the following taxonomy selections.

- Fasten → Tool → Right Angle Tool → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Apply Load →Maximum Grip

Approved for public release; distribution unlimited.
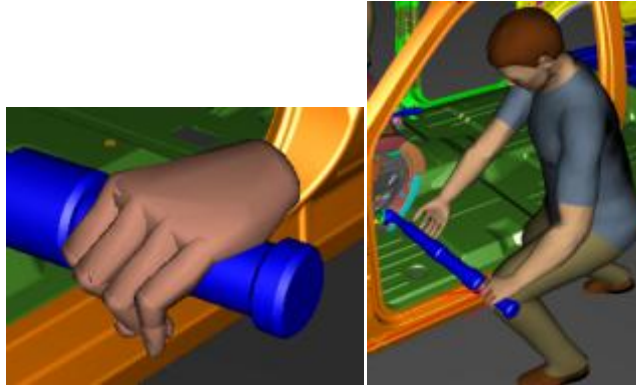
| 92 | Fasten | Tool | Right Angle Power Tool | Obtain, Align, Activate | Maximum Grip | move | | get | | move | | put | | move | | put | | load | |

Figure 69. Fasten IP on the Left Hand Side

And the third operator secures the IP to the dash at the front of the vehicle. First, a bolt is loaded using the following taxonomy selections.

- Assemble → Part or Assembly →Obtain, Gain Control, Align, Insert → Move, Get, Control, Move, Put, Apply Load →Five Finger Pinch Grip

| 7 | Assemble | Part or Assembly | Obtain, Gain Control, Align, Insert | Five Finger Pinch Grip | move | | get | | control | | move | | put | | move | | put | | load | |

Figure 70. Load Bolt for IP Fasten

And finally, secure the IP to the vehicle dash at the front of the vehicle using the following taxonomy selections.

- Fasten → Tool → Right Angle Tool → Obtain, Align, Activate → Move, Get, Move, Put, Move, Put, Apply Load →Maximum Grip
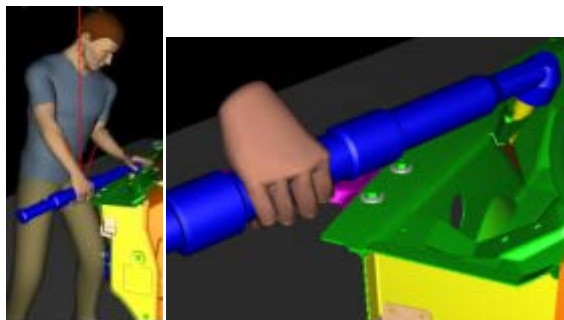
| 92 | Fasten | Tool | Right Angle Power Tool | Obtain, Align, Activate | Maximum Grip | move | | get | | move | | put | | move | | put | | load | |

Figure 71. IP Fasten at Dash in Front of Vehicle

4.1.2.3 Engine Dress

The heater hose installation operation includes obtaining the heater hose and installing it to the HVAC module port using the following taxonomy selections.

- Assemble → Part or Assembly → Obtain, Gain Control, Align, Press Fit → Move, Get, Control, Move, Put, Move, Put, Apply Load → Precision Grip
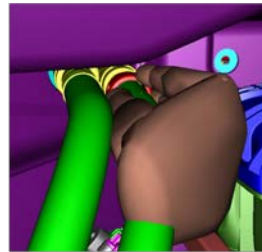
* Precision Grip

* Thumb Index Pinch Grip - more appropriate in this case
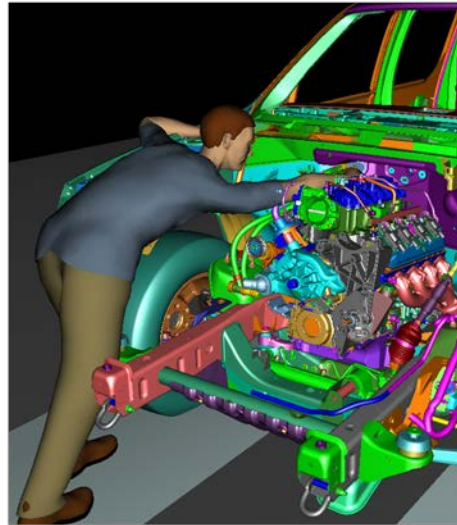
| 20 | Assemble | Part or Assembly | Obtain, Gain Contrl, Align, Press Fit | Full Hand Power Grip | move | get | control | move | put | move | put | load | |

Figure 72. Engine Dress Heater Hose Installation

The BEC module installation includes installing the BEC tray and securing electrical connections. The taxonomy selections are included below.

- BEC Tray Install - Assemble → Part or Assembly →Obtain, Gain Control, Align, Insert → Move, Get, Control, Move, Put, Apply Load →Five Finger Pinch Grip
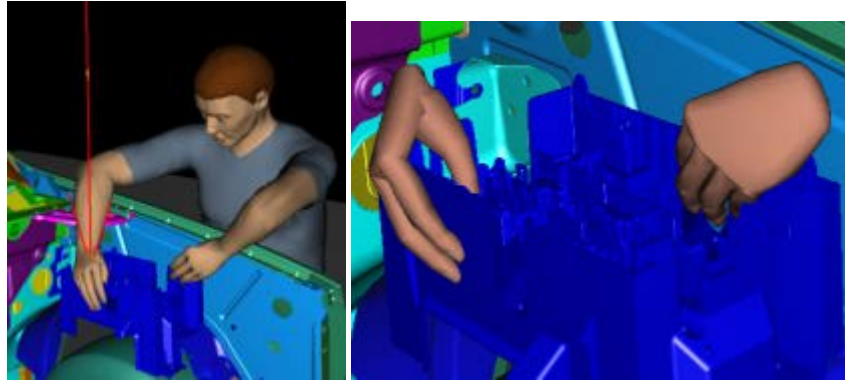
| 7 | Assemble | Part or Assembly | Obtain, Gain Control, Align, Insert | Five Finger Pinch Grip | move | | get | | control | move | | put | | move | | put | | load | |

Figure 73. Engine Dress BEC Tray Installation

- Secure Electrical Connections - Electrical Work → Part or Assembly → Obtain, Route, Connect → Move, Get, Move, Put, Move, Load, Move, Put, Load → Five Finger Pinch Grip
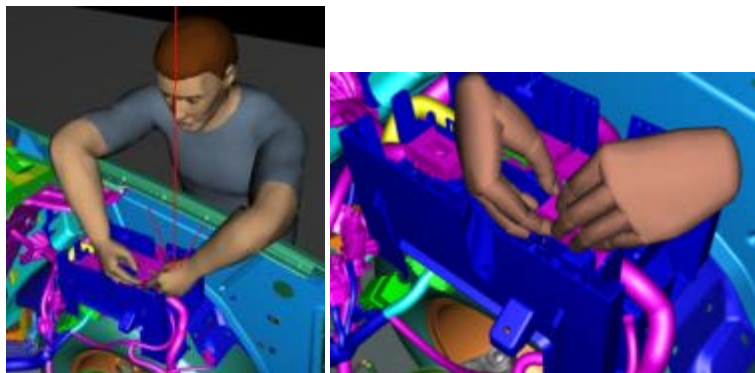
| 68 | Electrical work | Part or Assembly | Obtain, Route, Connect | Five Finger Pinch Grip | move | | get | | move | | put | | load | | move | | put | | load | |

Figure 74. Engine Dress Electrical Connections

- Secure Electrical Connections - Fasten→ Tool → Pistol Power Grip Tool →Obtain, Align, Activate→ Move, Get, Move, Put, Move, Put, Load → Precision Initial
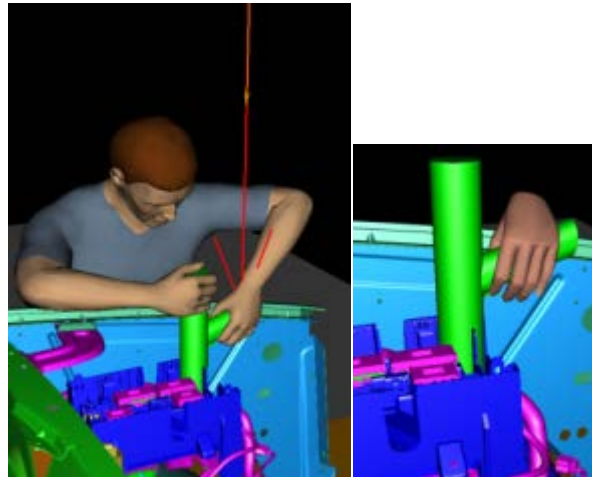
| 89 | Fasten | Tool | Pistol Grip Power Tool | Obtain, Align, Activate | Precision Initial | move | | get | | move | | put | | move | | put | | load | | |

Figure 75. Engine Dress Electrical Connections

4.2   Repetitive Work Assessment

One issue that arose in defining HAMML (and that is a significant factor in the design of processes for high volume manufacturing facilities) is the potential impact of repetitive work and corresponding fatigue. This section provides a review of current literature on repetitive work and fatigue assessment research that may impact iFAB assembly task planning. The literature search identified a number of research sources that are directly applicable in the iFAB domain. The identified research includes repetitive work threshold limits to minimize the risk of strain injuries for specific areas of the body and models to characterize work effort or fatigue.

Repetitive motion assessments and threshold values minimize musculoskeletal injury strains by flagging excessive repetitive motions localized to specific areas of the body. The most common focus areas for this research being the shoulder, arm and hand. Some examples are provided below in Section 4.2.1. A relatively simple computer implementation is expected for a couple reasons. The assessments require tracking movement of specific joints or segments of the body and most digital humans provide this functionality through a scripting or API interface, and minimal tracking of time is required. Due to the expectation of a relatively simple computer implementation, repetitive motion assessment methods are well suited to application in iFAB.

Work effort or fatigue assessments require significantly more detailed information in order to classify motions and tasks within work effort categories. In some cases, the assessments require calculation of the percentage of task cycle in high exertion. This category of assessment poses some implementation challenges within iFAB because the checks require detailed information for the assessment and the non-repetitive nature of iFAB work. Examples of this category of checks are provided in Section 4.2.2**.**

Approved for public release; distribution unlimited.

Section 3.1 provided a guideline for peak force for a given posture, specifying a "shall not exceed" threshold value. A similar consideration includes assessing the safe repetition rate for an operation or task may be characterized by a threshold value of X repetitions/hour or per shift or another time unit. The assessment should accurately characterize a worker in iFAB that is doing a number of different kinds of tasks (as opposed to a high volume automotive plant where workers basically do primarily the same set of tasks repeatedly through their shift). The specific work assigned to an operator should assess "safe repetition rates" or fatigue to drive decisions regarding "how many people are needed". In a sense, this iFAB system should factor in "rest" or "recovery" periods into a work-shift.

The following open questions remain unanswered: What is the best method to present and/or represent this information to the iFAB community? Given the "safe repetition rate" assessment requires size/weight of the parts, work time and other information including the work environment and layout, what is the process to reconcile needed information given the specific part and other information may not be known until the function is instantiated with a process plan, layout and other associated information. Given most of the assessments are impossible to validate accurately without knowing the actual worker posture and positions, and orientation of elements within the work environment, what is the best method to resolve the spatial questions?

4.2.1 Repetitive Motion Assessments

Repetitive motion guidelines are generally given for a specific area of the body, with the most common being shoulder, arm and hand. An example of limits for the finger/thumb and hand/arm is shown in Table 24.

Given 1) the type of exercise (dynamic or interim/static), 2) the weight of the load, 3) the risk level associated with the exertion 4) the body region required for the task (shoulder, upper arm, elbow, forearm, wrist, and finger) and 5) the number of repetitions required per time unit, Tables 24 and 25 provide examples of the guidance available for defining a threshold number of repetitions not to exceed for each body region required for the task. In addition, given a posture category (standing or stooped) and a load, similar thresholds are available for the number of repetitions not to exceed for lifting.

Table 24. Repetition guidelines for the finger/thumb and hand/arm.

Body Part	Minimum Force to Qualify as a Repetition	Repetition Trigger Value
Finger/Thumb	2 lb	>10 per minute
Hand/Arm	5 lb	>10 per minute

Table 25. Occupational Repetitive Action (OCRA) Posture Repetition Limits

The arms are not at or above shoulder level (flexion/abduction at 80° or more) for more than 10% of the cycle time and/or for more than 2 actions per minute.
The arms are not at mild abduction (between 45° and 80°) for more than one-third of the cycle time and/or for more than 10 actions/minute.
Sum total of wrist, elbow and grip actions should not exceed 30 actions/minute.

4.2.2 Fatigue (Work Effort)

Additional checks attempt to capture work effort by calculating duty time or percent of cycle time attributed to overall high exertion or high force work. This category of checks assesses repetitive work as a means to evaluate fatigue. Some analyses categorize work qualitatively based on motion speed, exertion level and frequency of effort, posing some challenges with respect to application through computer programs or virtual techniques. Some open questions include: How are "infrequent pauses" identified? What specific measures of digital human motion are associated with work "difficulty"?

The Strain Index is another tool used to evaluate a job's level of risk for developing a musculoskeletal disorder (MSD) of the hand, wrist, forearm or elbow [26][27]. The analyst evaluates six task variables including intensity of exertion, duration of exertion, exertions per minute, hand/wrist posture, speed of work, and duration of task per day. Three of the six task variables are subjectively evaluated by the analyst making them difficult to encode in an algorithm. The task variable is given a value called a multiplier. The multiplier values also pose iFAB implementation challenges because they are not explicitly defined, but instead are based on professional observations and opinions with support from physiological, biomechanical, and epidemiological principals as opposed to a mathematical relationship between task variables. The product of the six task variable multipliers produces a number called the Strain Index score. This score is used to identify the level of task risk. The Strain Index is limited to predicting risk for distal upper extremity musculoskeletal disorders. See Figure 76 for the formula. It was found that jobs associated with distal upper extremity disorders had scores greater than 5.

$$SI = (M_{Exertion\ Intensity})(M_{Exertion\ Duration})(M_{Exertions/Minute})(M_{Posture})(M_{Work\ Speed})(M_{Duration/Day})$$

Figure 76. Strain Index Formula

An example of Strain Index data is provided in Table 26. Note the smallest effort rating (1) requires a minimum of 2 lbs force exertion against an object. Motions without loading do not count.

Table 26. Strain Index data example

Effort Rating	Multiplier	Efforts per minute
1	0.5	< 4
2	1.0	4 to 8
3	1.5	9 to 14
4	2.0	15 to 19
5	3.0	≥ 20

There is a recent body of work was initiated in December of 2011 in the area of fatigue and recovery that is important to mention. The researchers involved include Jim Potvin (McMaster University), Tom Armstrong (University of Michigan), Laura Frey-Law (University of Iowa) and many others. The results of this effort are anticipated to be published by Jim Potvin at the Human Factors and Ergonomics Society Conference in 2012 with the possible title "*Predicting Maximum Acceptable Efforts for Repetitive Tasks: An Equation Based on Duty Cycle*". An eventual tool from this body of work might be a very good application for the iFAB project and would serve to plug an otherwise very difficult 'hole' in ergonomics analyses.

4.2.3   Usage

In the case of existing checks like hand clearance or reach, the trigger to perform an assessment is at the task level within the assembly taxonomy. Specifically if an operator reaches or places anything, perform a hand clearance or reach assessment.

For repetitive motion or fatigue assessments, the trigger to perform an assessment is at the operation level or some other time segment (like daily work assessment) to facilitate counting motion repetitions over a period of time. At each assembly task, the iFAB system could trigger an increase in repetitive motion counters for the appropriate motion (finger, hand, forearm, upperarm, elbow, shoulder or other body area or joint). However, the trigger for an assessment ("is the current counter value less than the appropriate threshold value?") should be triggered at the operation or other time segment level of assessment.

4.3   VME Implementation and Experimentation

The Virtual Manufacturing Environment provides a multi-user capable avatar based virtual world for prototyping a factory layout and manufacturing and assembly processes to be performed in the foundry. The VME layout and processes are automatically configured via XML input from the MCPML that includes a list of assets and resources required to perform the process as well as the detailed manufacturing capabilities. Users can modify the layout to

prototype virtual foundry configurations by directly interacting with the virtual environment. The environment supports avatar bots (computer controlled avatars) via scripting to perform foundry operations. Human controlled avatars may also interact with the foundry.

The hand grips and postures of both non-player "bot" avatars and human-controlled avatars update in real-time. The XML from the MCPML libraries specify which posture animation to play when an avatar is performing an assembly step within a process. The avatar reach zone (model based upon information from GM) can be graphically displayed or hidden (toggled on/off) by the user. Distance checks determine if an object is within the reachable zone of the avatar and failed attempts to reach objects out of the zone are logged as described below.

The Tundra platform was chosen for VME implementation because it satisfied the key requirements for an iFAB VME (Figure 77). It is a free open source system, satisfying AVMs requirement for open and extensible solutions. It provides a multi-user 3D virtual world allowing use of the VME for collaborative participation in foundry configuration and evaluation (and other VME use cases). It also includes real-time 3D object attachment or detachment, 3D real-time avatar posture animation, object or bot behavior scripting, collision detection, and the framework to generate 3D virtual objects in the environment from XML scene files.
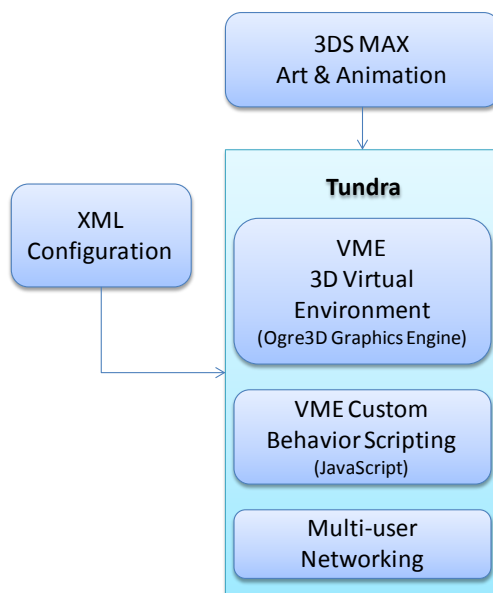
Figure 77. VME Architecture

4.3.1 Automatic Generation of Virtual Manufacturing Environment

At startup, the VME automatically loads the parts, tools, foundry structure, and 3D environment based upon the configuration file(s) that it loads. There are two types of files that configure the VME and will be automatically or semi-automatically generated for the VME based on setup. The factory scene file is an XML file following the Tundra scene schema and has the extension TXML. This file sets up the factory building and fixtures. For example, a crane or fixture location must be specified in the TXML configuration file prior to the server starting. When the VME is started in edit mode, the user can add, delete, or edit the fixtures and assets by directly

interacting with the environment. The user can save the new layout as a new TXML scene file for later use. External application can generate optimized layout in the TXML (read by the VME) to determine placement of cranes and other resources in an automatically generated factory layout.

Information specific to the particular assembly process is contained in an XML file created automatically from the database web services. This information consists of the tools, parts, and personnel required for the process tasks as well as the step-by-step actions for the process task. When the VME loads the MCPML file, the tools, parts, persons, are displayed in lists and the procedure steps are displayed in a table. 3D representations of the assets (tools, parts, and personnel) are automatically loaded and the VME guides the user to indicate where each asset is placed by simply clicking on a spot in the 3D environment.

The VME is designed to interface with MCPML library data through web services. The library interface has services defined to provide process and resource data to the VME. For a specified process, there is an XML file generated from that process to automate much of the configuration of the environment for the described operator, assemblies, tools, and other resources. Because the XML contains the work instructions for the assembly process, the MCPML XML format is loosely based on the S1000D international technical data specification for procedures. To generate each step with enough detail to automatically control the VME, information such as a particular type of hand posture and grip is specified for representation in the virtual environment. Figure 78 shows how the VME interfaces with the MCPML libraries.
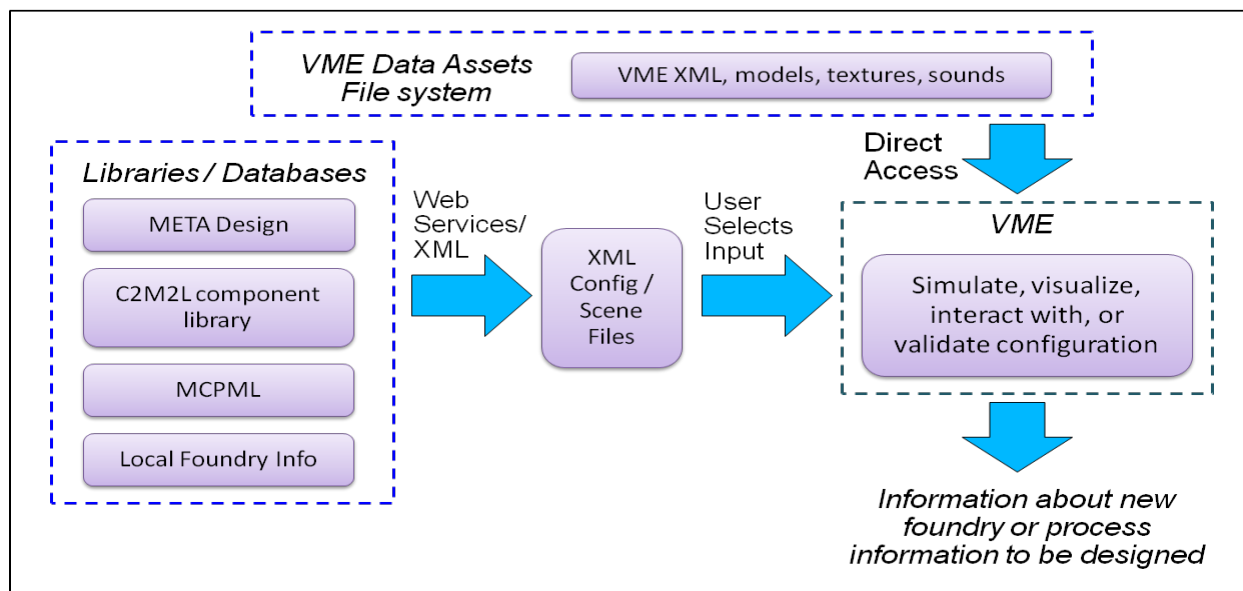
Figure 78. VME Interfaces

Once the MCPML is loaded and assets are placed, the user can step through the process script and view the tasks performed by the avatars as directed by the script. Each step is time-stamped and logged. Object collisions and failed attempts at reaching objects are automatically time-

Approved for public release; distribution unlimited.

stamped and logged with the steps. This is shown in the context of the integrated demonstration of Section 4.6 in Figure 92.

A particular TXML file could be generated from MCPML output restarting with a new TXML scene file. Therefore, it is recommended that the TXML scene file contains only the factory building and fixtures in the factory that will not depend on the MCPML files. The 3D VME environment and assets can be derived or converted from detailed CAD models. CAD files are not automatically converted to the 3D mesh models used by Tundra software.

4.3.2 Interacting with the Virtual Environment

Entity collisions are detected in the VME and reported. Typically in a multi-user virtual world, invisible simple collision geometries wrap around specific displayed objects to approximate collisions while maintaining reasonable simulation performance. Entities in the VME use boundary boxes rather than object geometry to detect collisions. Reach zones of the avatars are modeled using approximately spherical zones with origin placed at the avatar hip or shoulder. These reach zones are consistent with ergonomics work provided by General Motors. The VME identifies required assets that are outside the virtual operator's reach and logs the events.

Figure 79. Reachability Zones in the VME

Hand and torso postures are modeled in VME via real-time avatar animations. These are automatically triggered by in-program interface or from the process script file.

Approved for public release; distribution unlimited.
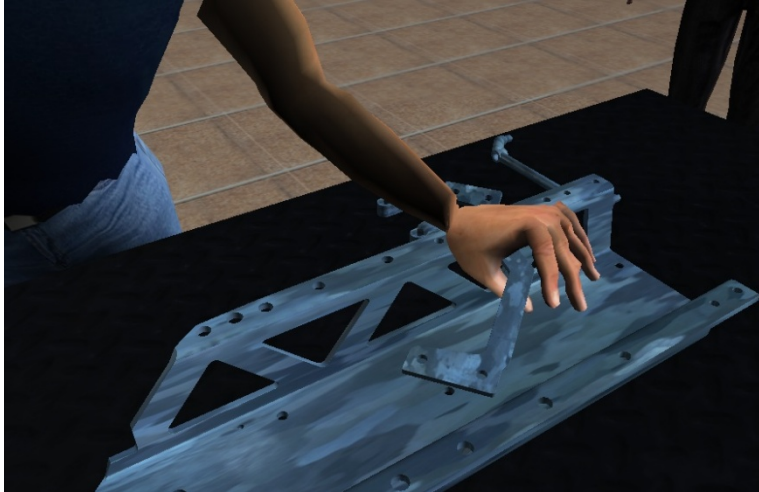
Figure 80. Hand Posture Modeling in VME

To use the VME, certain basic computer skills are required. Users need to know how to launch the application, maneuver through the VME menus, use the keyboard for simple avatar navigation, and use the mouse and keyboard controls to manipulate avatars, resources and control camera views in the 3D environment. Using the VME allows a user to experience and validate (via avatar interaction) the 3D environment and manufacturing process for a user selected foundry configuration which is loaded from configuration files.

4.4   MCPML Population and Web Service Interface

The Boeing iFAB MCPML is implemented as a suite of web services supported by a database backend (Figure 81). The web services are divided into four categories: manufacturing capability, library update, design specific fabrication, and design specific assembly. The backend database is divided into two SQL databases. MCPML_PROC_V2 contains the HAMML process data. MCPML_V2 contains data for other foundry resources, including machines, tools, and tooling. The library also includes external resources for data that does not fit within the database. For the final iFAB release, this consists of graphics files for various HAMML hand and body postures.

Approved for public release; distribution unlimited.
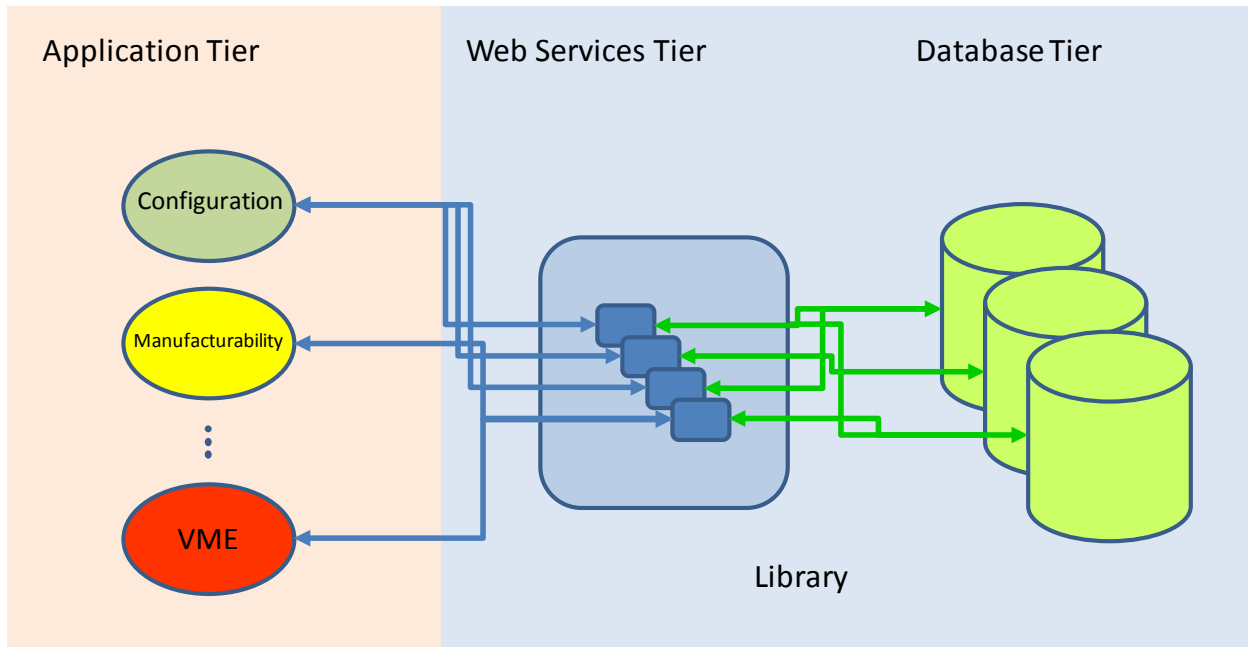
Figure 81. MCPML Implementation Architecture

The final iFAB release of the library contains almost 2000 entries of various types, and over 45 web services. Though this is the final iFAB release, the Boeing MCPML is being further extended and populated under C2M2L-1, and future C2M2L-1 releases may include additional external resources where needed, such as CAD or mesh files for tools or machines. The current iFAB releases also includes the MCPML_BOP_V2 database containing Bill of Process elements largely populated by the Boeing C2M2L-1 activity.

Though the raw data provided by the library may be retrieved by accessing the databases directly, primary access to the library is through a suite of web services. The web services provided by the MCPML can be organized in four categories.

The first category, manufacturing capability web services, provides essentially raw access to the data in the various MCPML databases. These services typically come in pairs, one that lists all of the elements in the library of a certain type (e.g., machines, hand tools), and a second that provides detailed information about a specific element in the library. For example, `getProcessNames` retrieves the list of HAMML processes, while `getProcessSpecifications` takes as input a process name and returns the details of the process. The second category of web services allows updating of the library via the web service interface. In particular, there are web services for adding and deleting various library elements, as well as services for getting library identifiers for library elements. Since these services can directly affect the data in the library, their use should be limited to authorized users (such as the library curator), and care should be taken since the effects of the services cannot be undone.

The third and fourth categories of services implement business logic related to design specific fabrication and assembly activities respectively. An example of the third category is

`capableMachinesWorkpiece`, which determines the machines that can perform a particular fabrication process on a given workpiece, specified by its bounding box and weight. An example of the fourth category is `getAssemblyTimeForFastenProcess`, which determines the time required to assemble a pair of parts according to a given fastening process.

The library is being populated from data captured in our work from characterization of machines, tools, factories, assembly processes, and human actors in earlier tasks of our statement of work. The specific machines, tools, and processes that constitute the initial library have been identified as those with relevant process coverage of a foundry capable of producing heavy infantry fighting vehicle drivetrains. Also, library entries are created with appropriate factory-level information and human element models.

This library has been shared with other iFAB, META, and AVM performers through a suite of web services. The web services allow access of the database information across the internet when deployed on a host server with appropriate internet access. The library is also deployable locally and may be accessed via a local interface.

The specifics of the web services, including pseudo-APIs and example SOAP 1.2 request and reply messages are provided in the "iFAB MCPML Interface Description" document, included with the MCPML distribution and appearing as Appendix E of this report. The MCPML distribution also includes the WSDL file giving the formal definitions of the services, as well as the service implementations, the MCPML database structure and contents, and a README file providing instructions for installation and use.

4.5   Using HAMML in Foundry Configuration

This report has focused on describing the philosophy, design, and capabilities of the Boeing MCPML. Though designed to support the various AVM use cases, including foundry configuration and design manufacturability analysis, the most explicit reference to this point has been the description of the motivating questions presented in Appendix E that were used to design the MCPML web services. In this section, we examine a particular MCPML use case, the generation of assembly instructions during foundry configuration.

Consider a conceptual model of assembly instructions represented as a tree diagram (Figure 82) with joining nodes representing assembly tasks which join parts and sub-assemblies together. Leaf nodes represent parts and sub-assemblies. The tree defines the "as built" subassemblies. A post order traversal of the tree (where the subtrees of a node are visited before the node itself) would then represent an "assembly sequence".

Approved for public release; distribution unlimited.
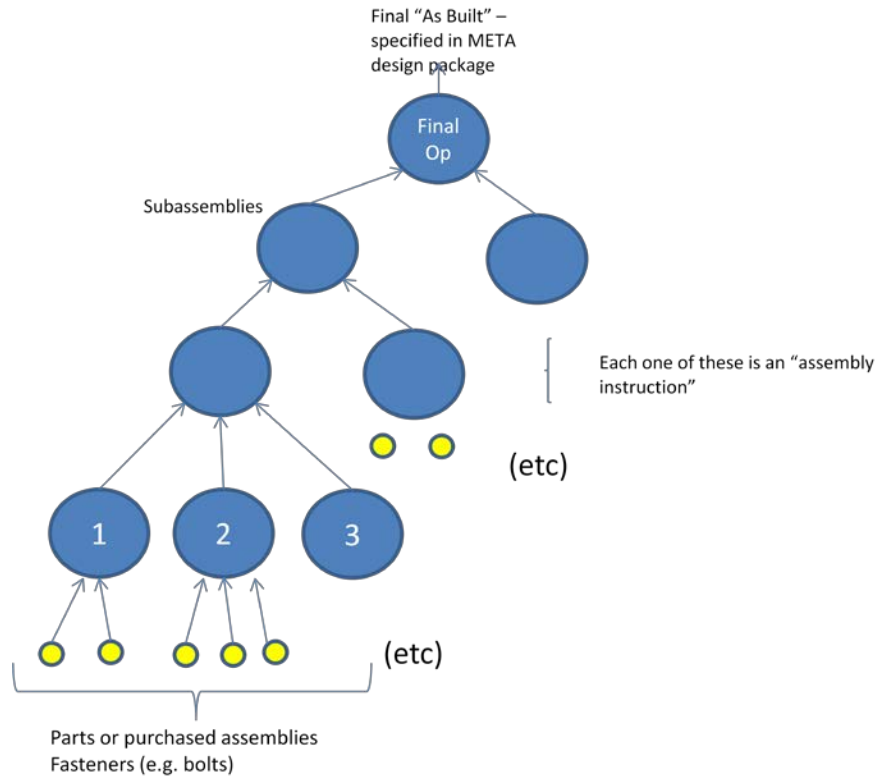
Figure 82. Assembly Instruction Tree Diagram

We propose defining a "mechanical fastening operation" (Figure 83) as something that joins 2 or more parts. This yields an Assembly Instruction Tree Diagram similar to a liaison graph. However this graph must represent any order of operation constraints, including concurrency

First we consider what must be included in the "operation" nodes. The type of joining method must be represented. Examples include bolt, screw, weld and glue. Additional information will be included depending on the joining method. For example, if it is "bolt" then additional assembly information includes the actual bolt geometry, where they are positioned within the assembly and the number of them. Torque must also be represented for each bolt as the value might not be the same for each bolt. The simplest assumption is the bolts themselves are parts (or part instances, i.e. in the CAD package the geometry can be represented once, and then an "instance" of it is created by providing a coordinate system indicating where it is located). They are represented in the tree at the operation node.

The TDP needs to include this information (joining method, actual bolts, and specification). It also needs to include the relationship that says "this fastener connects these parts". If this information is not explicitly included in the design package, then "spatial reasoning" is needed to figure out based on the geometry, the joining method and all the related information. This is not only difficult (and probably even ambiguous), but is also unnecessary and error prone. The designer knows what parts are being bolted.

Approved for public release; distribution unlimited.

Given the design, the following is a conceptual algorithm for generating the assembly instruction tree. Once the tree is generated, a specific sequence can be generated by a post order tree traversal with appropriate orderings of subtree visits. Note that MCPML/HAMML library elements, tools, and services have been defined to realize many of the steps of the proposed algorithm (or to provide the appropriate manufacturing capability information for some other algorithm to use to implement the step).

- Start with a "forest" of singleton nodes corresponding to parts or purchased assemblies.
- Pick a "fastener" that has not yet been put into a tree. Possibly the right word to use in META context is "structural interface definition".
- For each part that the fastener fastens (i.e. query the fastening relation), find the root nodes of the trees in the current forest containing those parts.
- Create a new node, attach the fastener to it, and the nodes identified above.
- Optionally, attach any other unattached fasteners whose associated parts are contained entirely below the newly created node.
- Continue doing this until all fasteners (or structural interface definitions) have been attached. Note that if after doing this there is NOT a single tree, we can say that the product assembly package is incomplete.
- Verify that there is a collision free path for bringing parts and fasteners into position.
- Verify that there a process or tool in our library capable of performing the task and meeting the specification of the operation.
- Identify any additional tools, fixtures, or other resources required. NOTE: the tool, extensions, fixtures, etc, also need a collision check. Conceptually this involves ADDING models for tooling/etc to this tree that corresponds to the resources needed for the operation. These plant elements must also include the position of the tool, fixture, and the motion of the tool as the operation is performed. Generating some of the additional needed information may be non-trivial.
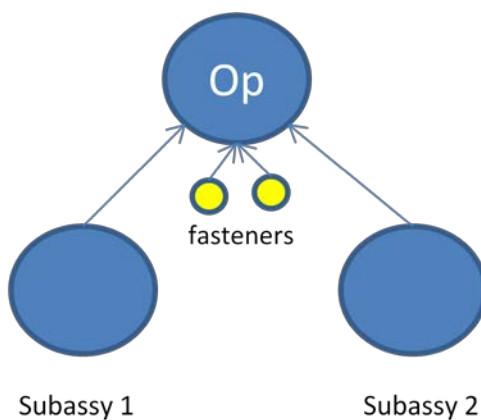
Figure 83. Mechanical Fastening Operation

The next section describes an integrated demonstration of the MCPML HAMML capabilities using just such an algorithm.

Approved for public release; distribution unlimited.

4.6 Integrated Demonstration

Boeing worked with other AVM contractors to conduct an experiment to integrate and demonstrate the language and libraries developed for iFAB. This experiment integrated the Boeing Manufacturing Capability and Process Model Library, Boeing Human Assisted manufacturing Model Library, CMU Motion Planning API, Georgia Tech MLibrary, Penn State ARL Liaison Graph, Vanderbilt Meta Assembly BOM and the Boeing Virtual Manufacturing Environment.
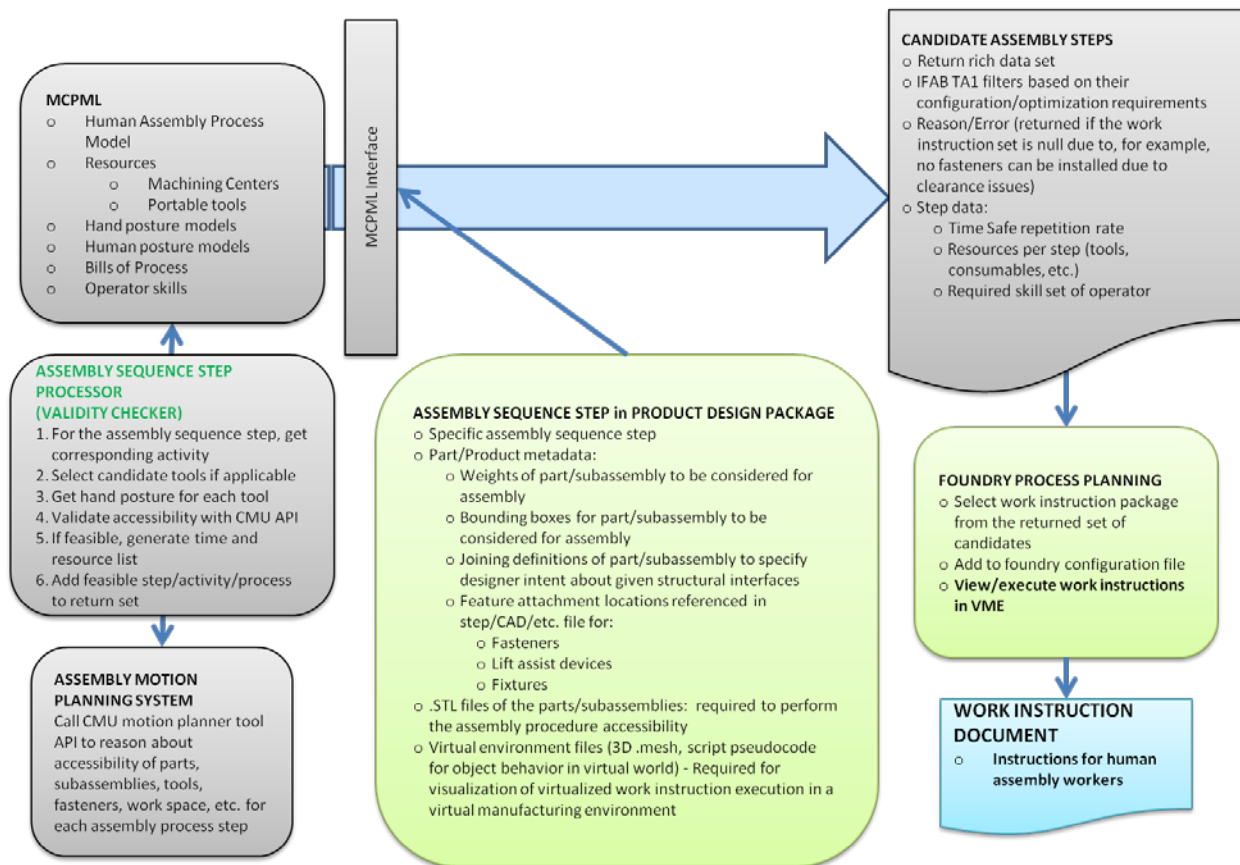
Figure 84. Experiment Flow

The experiment took a representative META design and from it created candidate assembly sequences. The candidate assembly sequences were then tested for manufacturability by using the tools and libraries created by AVM performers. Once a validated sequence was created it was demonstrated in the Virtual Manufacturing Environment and assembled by a human.

Approved for public release; distribution unlimited.
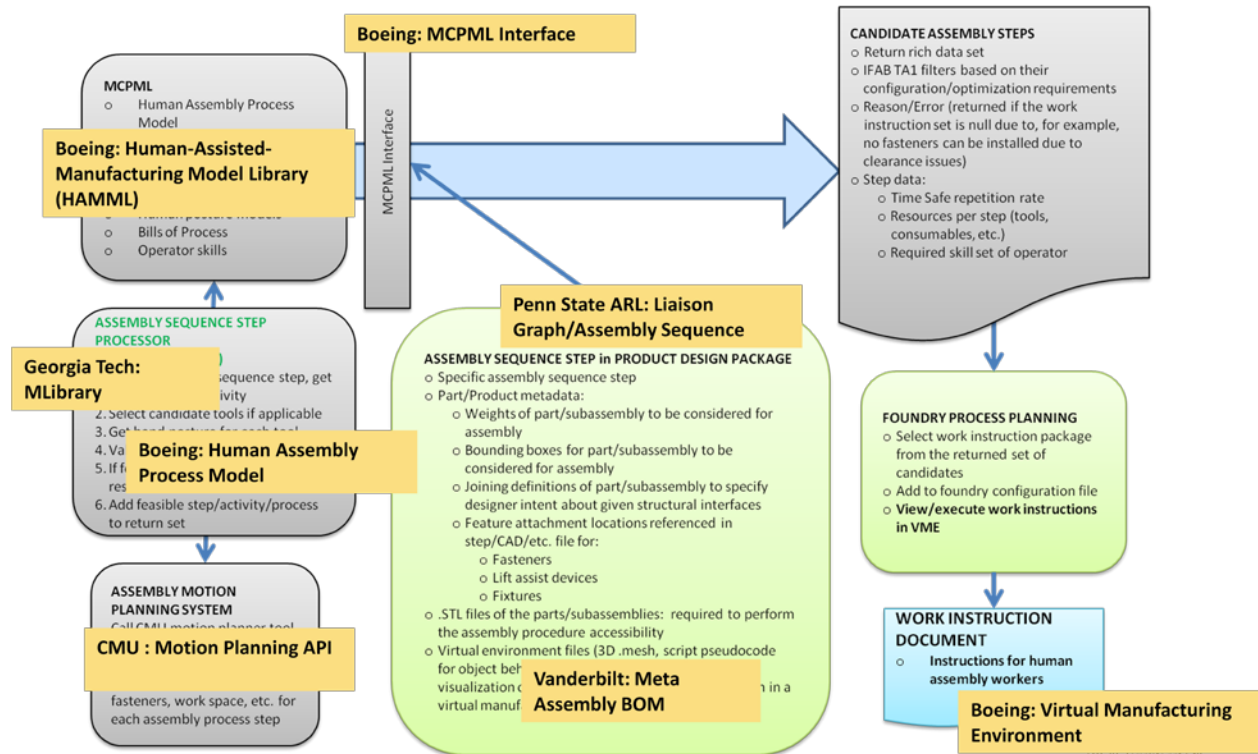
Figure 85. Experiment Flow with AVM Mappings

4.6.1 AVM Integration

The Vanderbilt Meta Assembly BOM contains information about parts and fasteners, materials, sizes, weights and CAD file names. It was used as a source of input information when a candidate assembly sequence was validated using the MCPML interface.

The Penn State ARL: Liaison graph refers to parts in an assembly via GUIDS. It also contains a mapping of GUIDS to STP file names. The STP file names can be parsed to retrieve a part number which can be used to retrieve part information from the Vanderbilt Meta Assembly BOM. The Liaison graph can be used to create a candidate assembly sequence for an assembly.

The Georgia Tech MLibrary provides information on candidate tools for particular fasteners. When a candidate assembly sequence is created it can then be parsed for fasteners and the MLibrary used to select correct tools to use on the fasteners.

The Boeing Human Assisted Manufacturing Model Library contains model information on human assembly processes. Each step in the candidate assembly sequence is input into the MCPML interface and the human assembly process steps are returned if the step is valid. Information about cost and time are also returned.

Approved for public release; distribution unlimited.

The CMU Motion Planning API verifies if there is an unobstructed path to join the parts. It is also used to verify that a hand and tool can fit in the position necessary to fasten the parts together.

The Boeing Virtual Manufacturing Environment takes the validated candidate assembly sequence and displays the assembly sequence in a virtual factory.

4.6.2   Experiment Assembly Selection

The Meta Challenge Problem was selected to validate the integration.
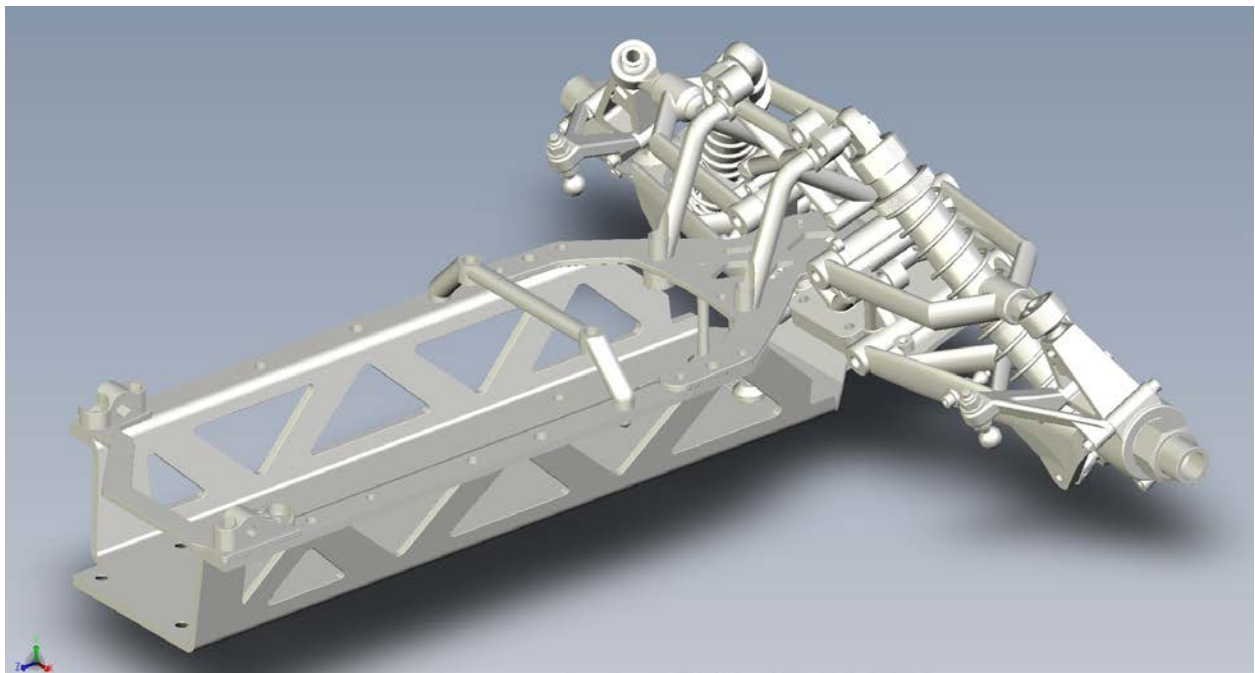
Figure 86. Meta Challenge Problem

Approved for public release; distribution unlimited.

4.6.2.1 Selection of Sub-Assembly

The assembly was examined to find a suitable sub-assembly to perform the experiment on. The right front suspension assembly was selected because it had a number of fasteners that might be difficult to reach depending on the selected assembly sequence.

A DDM model was created of the suspension to verify the manufacturability of the candidate sequence that was selected by the automated manufacturing process (Figure 87).
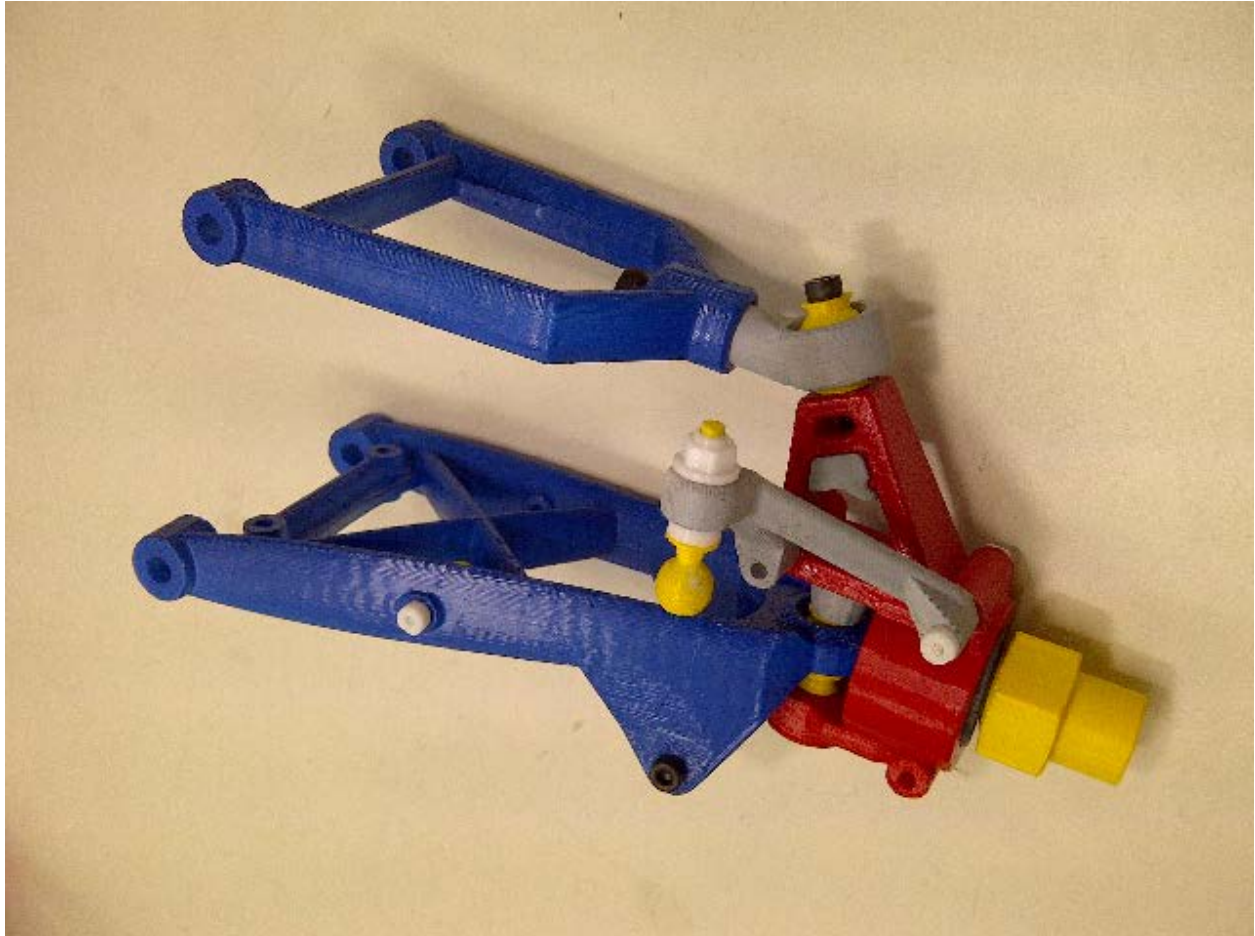
Figure 87. DDM model of the Right Front Suspension

Approved for public release; distribution unlimited.

4.6.3 Experiment Description

Create a candidate assembly sequence from the liaison graph.

Tools selected based on the fasteners.

Assembly sequence validated against the Manufacturing Capability and Process Model Library.

Detailed work instructions are generated.

Detailed work instructions are executed in the Virtual Manufacturing Environment.

Detailed work instructions validated by assembling the DDM model by hand.

4.6.3.1 Assembly Steps Constructed from Liaison

The first step in constructing a candidate assembly sequence was interpreting the liaison graph. META information is found for each entry in the liaison graph by mapping GUIDS to STP file names. The STP file names are then parsed for part numbers which are then used to find the CAD file name in the META BOM. By identifying the parts and fasteners, candidate assembly sequences were generated.

Assembly	Notes	Part 1	Part 2	Fastener	Number of Fastene	Type	x1	y1	z1	w1	x2	y2	z2	w2	Tools
Front_arm_lower_right.asm		86417	85422_6			press	1	1	1	0.02	1	1	1	0.02	
Front_arm_lower_right.asm		85400_2	85422_6			press	4	3	1	0.02	2	1	1	0.02	
Front_arm_lower_right.asm		85400_2	86405			loose	4	3	1	0.3	1	0.5	0.5	0.02	
Front_arm_lower_right.asm		85400_2	86405	z216		fasten	4	3	1	0.3	1	0.5	0.5	0.02	Allen Wrench
Front_arm_lower_right.asm		85400_2	86405			fasten	4	3	1	0.3	1	0.5	0.5	0.02	Speed (open end) Wrench
Front_arm_upper_right.asm		85400_5	85400_1			insert	1	1	0.25	0.02	4	3	1	0.3	
Front_arm_upper_right.asm		85400_1	85416_6			insert	4	3	1	0.3	2	1	0.5	0.05	
Front_arm_upper_right.asm		85400_1	85416_6	94909		fasten	4	3	1	0.3	2	1	0.5	0.05	Allen Wrench
Front_arm_upper_right.asm		85400_1	85400_3			loose	4	3	1	0.3	1	1	0.25	0.002	
Front_arm_upper_right.asm		85400_1	85400_3			loose	4	3	1	0.3	1	1	0.25	0.002	
Front_hub_right.asm		85408	85408_1			loose	3	2	2	0.4	2	2	2	0.25	
Front_hub_right.asm	Assemble bolt to washer	z303	z224			loose	3	0.5	0.5	0.01	0.5	0.5	0.025	0.01	
Front_hub_right.asm	Assemble bolt to washer	z303	z224			loose	3	0.5	0.5	0.01	0.5	0.5	0.025	0.01	
Front_hub_right.asm	Assemble washer and nut	z663	z224			loose	0.5	0.5	0.025	0.01	0.5	0.5	0.025	0.01	
Front_hub_right.asm	Assemble washer and nut	z663	z224			loose	0.5	0.5	0.025	0.01	0.5	0.5	0.025	0.01	
Front_hub_right.asm	Put parts together	85408	85408_1			loose	3	2	2	0.4	2	2	2	0.25	
Front_hub_right.asm	Insert bolt	85408	z303			insert	3	2	2	0.4	3	0.5	0.5	0.01	
Front_hub_right.asm	Insert bolt	85408	z303			insert	3	2	2	0.4	3	0.5	0.5	0.01	
Front_hub_right.asm	loose assemble nut	85408_1	z663			loose	2	2	2	0.25	0.5	0.5	0.025	0.01	
Front_hub_right.asm	loose assemble nut	85408_1	z663			loose	2	2	2	0.25	0.5	0.5	0.025	0.01	
Front_hub_right.asm	tighten bolt	85408	85408_1	z303	2	fasten	3	2	2	0.4	2	2	2	0.25	Allen Wrench
Front_hub_right.asm	tighten nut	85408	85408_1	z663	2	fasten	3	2	2	0.4	2	2	2	0.25	Speed (open end) Wrench
Front_hub_right.asm	put big washer on	105612	b089			insert	0.8	0.8	1.7	0.05	1	1	0.25	0.25	
Front_hub_right.asm	put big washer on	105612	b089			insert	0.8	0.8	1.7	0.05	1	1	0.25	0.25	
Front_hub_right.asm		85408	105612			loose	3	2	2	0.4	0.8	0.8	1.7	0.05	
Front_hub_right.asm	assemble bolt and washer	94510	105613			loose	3	0.5	0.5	0.01	0.5	0.5	0.025	0.01	
Front_hub_right.asm	assemble bolt and washer	94510	96704			loose	3	0.5	0.5	0.01	0.5	0.5	0.025	0.01	
Front_hub_right.asm		85408	105612	94510	1	fasten	3	2	2	0.4	0.8	0.8	1.7	0.05	Allen Wrench
Front_hub_right.asm		85408	85422_3			press	3	2	2	0.4	0.5	0.5	0.5	0.02	
Front_hub_right.asm		85408	86425			insert	3	2	2	0.4	0.5	0.5	0.5	0.02	
Front_hub_right.asm		86412	85422_8			insert	1.3	0.2	0.2	0.01	1	1	0.5	0.02	
Front_hub_right.asm		86412	85408_1			insert	1.3	0.2	0.2	0.01	2	2	2	0.25	
Front_hub_right.asm		86412	z682			loose	1.3	0.2	0.2	0.01	2	2	2	0.25	
Front_hub_right.asm		86412	85408_1	z682	1	fasten	1.3	0.2	0.2	0.01	2	2	2	0.25	Speed (open end) Wrench
Front_hub_right.asm		85408	86425			insert	3	2	2	0.4	0.5	0.5	0.5	0.02	
Front_hub_right.asm		85408	85408_2			insert	3	2	2	0.4	0.5	0.5	0.5	0.02	

Figure 88. A Candidate Assembly Sequence with Fasteners Identified

4.6.3.2 Tool Selection Based on Fasteners

The candidate assembly sequence was then parsed for fasteners and the MLibrary was called to select tools.

```
=== Results for Query 1a
Types of fastening tools that can be used for MetricSocketHeadCapScrew
...Hex Key (id = ec8f5507-f560-41da-8fbb-d83703d7ec27)
...Hex Bit (id = 2d2babb1-12d3-4077-bee6-ad27448ba4d7)
...Hex Socket Bit (id = 2a0be7c1-f0e0-4bbb-8e09-59005491a136)

=== Results for Query 1b

Hex Key is a manual hand held fastening tool that does not require a machine

Machines that can be used for the tool: Hex Bit
...Non-Graduated Screwdriver Grip Limiting Torque Wrench with Internal Hex Drive (id = 822c4c33-3969-4c45-8322-e5ea5b8c71a3)
...Graduated Screwdriver Grip Limiting Torque Wrench with Internal Hex Drive (id = c69ee044-cf5d-42b1-9bbc-67e8c4f9b426)
...Multi-Bit Screw or Nut Driver Handle (id = 1b342b29-d091-43c4-9f89-0fb7ef6580d2)
...Electric Screwdriver (id = 7b0878a3-8444-4d8e-9216-805d56e5d637)

Machines that can be used for the tool: Hex Socket Bit
...Sliding T-Handle (id = 4872b1dc-282d-4f0d-8cda-5bc6435006a8)
...Electric Impact Wrench (id = ae137068-9fe2-4a20-9ff4-041f3cdbbf52)
...Hinged Handle (id = 6dec7bed-d041-419e-a21f-8f346a5d95a1)
...Pneumatic Impact Wrench (id = fc677d86-70bc-4492-8d0b-5507300610a1)
...Spin Type Screwdriver Grip Speeder Handle (id = dce87888-68ad-446c-a70f-3736c4f38fa1)
...Reversible Ratchet Handle (id = 7fe30830-867b-46ff-87c2-86e5034fad64)
...Electric Ratchet Wrench (id = 7adcfed0-e1e7-4e36-92b0-48be96cbca93)
...Torque Wrench (id = de9f40e1-2524-4b1d-9b4f-a555073444c8)
...Brace Type, Single Revolving Hand Grip Speeder Handle (id = c943ef4f-276a-46e0-9195-f3b7d1df99f1)
...Pneumatic Ratchet Wrench (id = d5f6cb4b-d79e-40f1-bf49-755e7b3c79dc)
************************************************************************************************
```

Figure 89. MLibrary output for Metric Socket Head Cap Screw

4.6.3.3 Candidate Assembly Sequence Validated

The assembly steps were checked for accessibility with the CMU Motion Planning API.
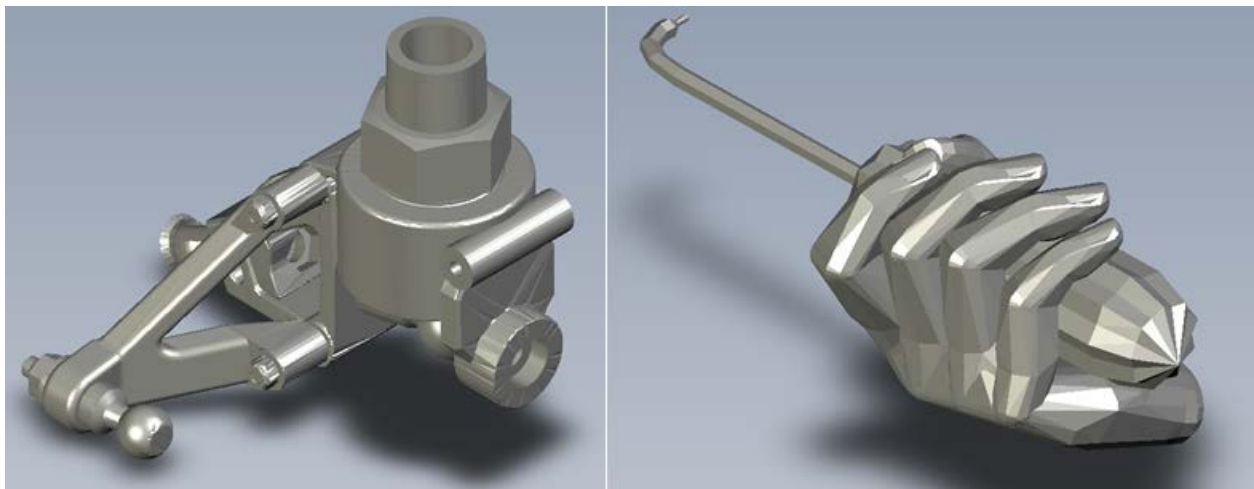
Figure 90. Accessibility Checks

4.6.3.4 Detailed Work Instructions Generated

The Human Assisted Manufacturing Model Library was used to verify the manufacturability and generate process steps/time, resources needed and work instructions.

```xml
<?xml version="1.0" encoding="utf-8" ?>
- <WorkInstructions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema":
  - <steps>
    - <AssemblyStep>
      - <assemblers>
          <string>Person1</string>
        </assemblers>
      - <part1>
          <number>86417</number>
          <name>BALL 10x12mm</name>
        </part1>
      - <part2>
          <number>85422_6</number>
          <name>NUT HOLDER SET 6</name>
        </part2>
        <fastener />
      - <process>
        - <AssemblyProcess>
            <executeTime_minutes xmlns="http://ifab.boeing.com/">0.02795</executeTime_minutes>
          - <parts xmlns="http://ifab.boeing.com/">
            - <Part>
                <xDimension_in>1</xDimension_in>
                <yDimension_in>1</yDimension_in>
                <zDimension_in>1</zDimension_in>
                <weight_in_lb>0.02</weight_in_lb>
              </Part>
            - <Part>
                <xDimension_in>1</xDimension_in>
                <yDimension_in>1</yDimension_in>
                <zDimension_in>1</zDimension_in>
                <weight_in_lb>0.02</weight_in_lb>
              </Part>
            </parts>
          - <fasteners xmlns="http://ifab.boeing.com/">
            - <Fastener>
                <Method />
                <Medium />
                <Count>0</Count>
              </Fastener>
            </fasteners>
            <resources xmlns="http://ifab.boeing.com/" />
          - <sequence xmlns="http://ifab.boeing.com/">
            - <ProcessActivity>
                <name>Assemble</name>
                <activityId>1002</activityId>
              - <taskSets>
                - <StepList>
                    <objectCategory>Part or Assembly</objectCategory>
                    <execute_minutes>0.02795</execute_minutes>
                    <taskSetId>1000</taskSetId>
                  - <steps>
                    - <TaskStep>
```

Figure 91. Example of Rich Dataset Work Instructions

Approved for public release; distribution unlimited.

4.6.3.5 Work Instruction Execution in the Virtual Manufacturing Environment

The Virtual Manufacturing Environment was used to view and execute the work instructions. Using it verified the work instructions and can be used to manage foundry configuration.
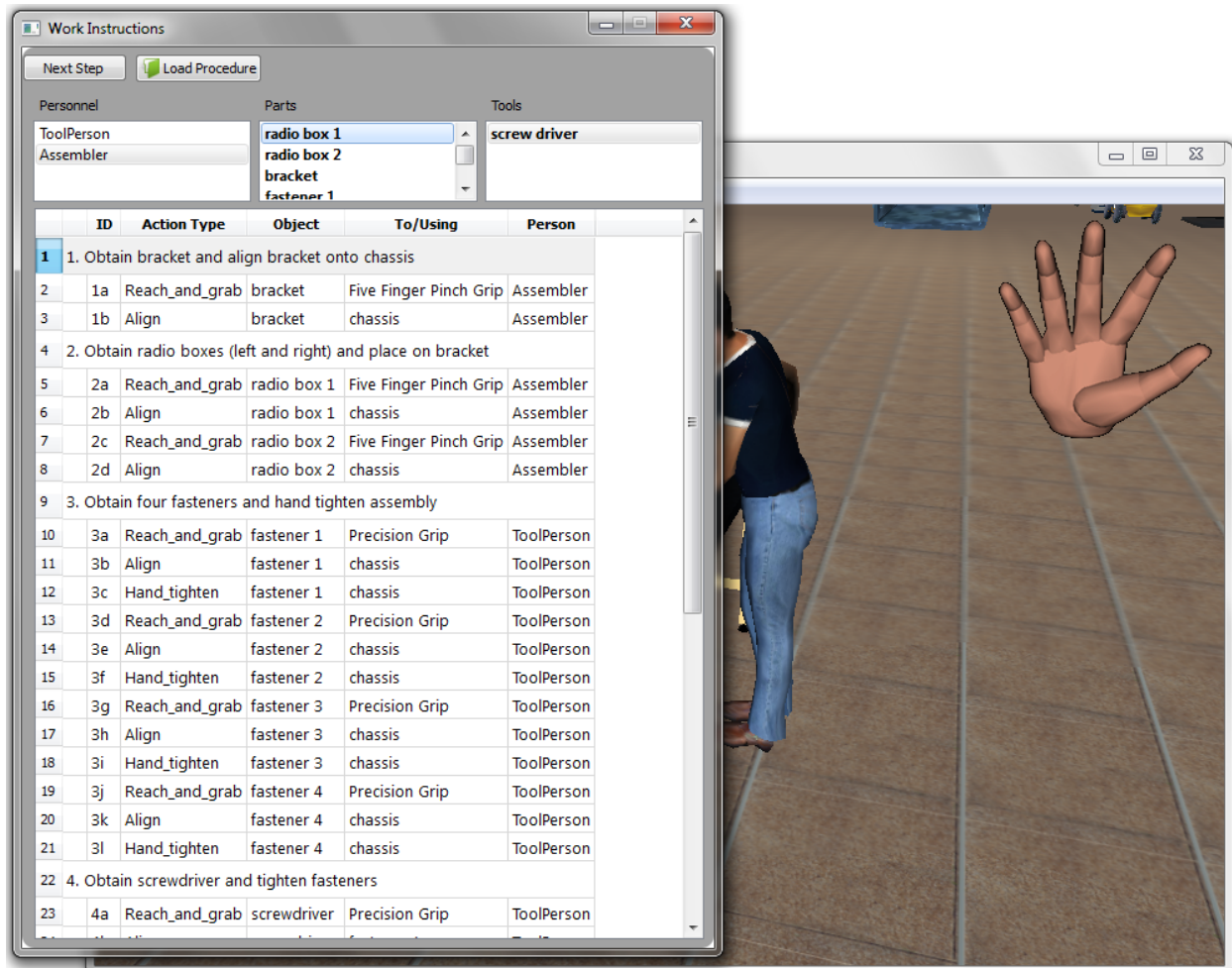
Figure 92. Work Instruction Execution in the VME

4.6.3.6 Human Assembly of DDM Suspension Model

The final step in the experiment was to validate the work instructions by assembling the DDM model using generated detailed work instructions.

4.6.4   Conclusion and Results

Boeing was able to verify via integrated experimentation the viability of the AVM products that Boeing developed for iFAB:

Human Assembly Process Model

Human-Assisted-Manufacturing Model Library (HAMML)

MCPML Interface

Virtual Manufacturing Environment

Integrated Experimentation.

Boeing successfully created detailed human assembly work instructions that could be visualized in a virtual environment or used by a human to assemble a sub-assembly specified by a META generated design.

5.0   CONCLUSIONS

The instant Foundry Adaptive through Bits (iFAB) relies on a manufacturing model library as the datastore for process and capability information. This report discussed the implementation of the Boeing developed manufacturing library and its ancillary components.

A manufacturing model language was created by characterizing aspects of a manufacturing facility alternatively referred to as a foundry. The foundry aspects are identified as machines, tools, material handling equipment, and other tangible resources. Infrastructure and facilities details are characterized. The human actor of manufacturing is captured with respect to actions and effects the human actor has on fabrication and assembly activities. All of these foundry features and elements are semantically defined in a manufacturing capability and process model language.

The aforementioned manufacturing language defines a library with data about machines, tools, processes, resources, human activities, aggregated process descriptions, and more. This Manufacturing Capability and Process Model Library is presented to external tools via an interface. That interface to the MCPML contains the logic and heuristics needed to place the manufacturing elements and resources into appropriate context for manufacturing processes. The MCPML interface provides data to queries for manufacturability, process sequencing, and foundry configuration. The interface is extensible and service-oriented.

In addition, a prototype Virtual Manufacturing Environment (VME) was created to ascertain manufacturability from a crowd-sourced perspective and virtual configuration of the iFAB foundry. VME uses an open-source "virtual worlds" engine. Assembly process information draws from the MCPML interface as XML data which shows, through use of avatars, manufacturing feedback on reachability, spatial configuration, part and work cell layout, and potential for collaboration between multiple users. The VME prototype interfaces with the MCPML library data through web services. The library interface has services defined to provide process and resource data to the VME. For a specified process, there is an XML file generated from that process to automate much of the configuration of the environment for the described operator, assemblies, tools, and other resources.

Boeing analyzed a number of assembly operations for human process capability using the human assembly process model. The operations analyzed include the iFAB Meta example and the GM engine marriage, instrument panel installation and engine dress.

Each element in the Human Assembly Process Model taxonomy gives detailed information about a manufacturing task, provided that some high level information (Activity, Object, Plant Equipment, Material or Tool) is known about the manufacturing process.   To use the taxonomy, information about the assembly operations is used to narrow down the search for task definition by specifying information in each column of the taxonomy.

Boeing demonstrated the utility of the MCPML at the May PI meeting in an integrated demonstration highlighting the use of the MCPML capability models in conjunction with design elements from a META TDP and a liaison graph generated from the TDP by Penn State ARL algorithms. These were used to define a candidate manufacturing sequence, and each step in the

sequence was checked for feasibility with respect to the MCPML foundry capabilities models, including the use of motion planning algorithms developed by CMU to check that the proposed operations did not violate product and tool geometric constraints. The VME was also used to visualize the resulting plan which would allow foundry operators and other SMEs to perform qualitative validation of the plan. This validated both the capability models and other facilities provided by the MCPML and VME, but also showed a proof of concept integration of a non-trivial subset of the overall iFAB toolchain required for iFAB Foundry success.

6.0 REFERENCES

[1] American Society of Mechanical Engineers. (1994). ANSI / ASME Y14.5.1M-1994 Standard on Dimensioning and Tolerancing. ANSI, New York.

[2] ASME Y14.5-1994 (1999). Dimensioning and tolerancing. ASME.

[3] Hong, Y. S., & Chang, T. C. (2003). Tolerancing algebra: a building block for handling tolerance interactions in design and manufacturing. Part 2: Tolerance interaction. International journal of production research, 41(1), 47-63.

[4] Hong, Y. S., & Chang, T.-C. (2002). A Comprehensive review of tolerancing research. International Journal of Production Research, 40(11), 2425-2460.

[5] Hong, Y. S. and Chang, T. -C. (2002). Tolerancing algebra: A building block for handling tolerance interactions in design and manufacturing Part 1: Concept and representation. International Journal of Production Research, 40: 18, 4633 - 4649.

[6] Huang, S. H., Liu, Q., & Musa, R. (2004). Tolerance-based process plan evaluation using Monte Carlo simulation. International journal of production research, 42(23), 4871-4891.

[7] Groover M.P. (1996). Fundamentals of model manufacturing. Prentice Hall, Upper Saddle River, NJ (now published by John Wiley, New York), pp 634.

[8] Kalpakjian, S., & Schmid, S.R. (2006). Manufacturing engineering and technology. Pearson Prentice Hall, Upper Saddle River, NJ, pp. 607-646.

[9] Oberg, E., Jones, F. D., Horton, H. L., & Ryffel, H. H. (2008). Machinery's Handbook (28th Edition) & Guide to Machinery's Handbook. Industrial Press, pp 715. Online version available at: http://www.knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid =2422&VerticalID=0

[10] Okafor, A. C. & Ertekin, Y. (2000). Derivation of Machine Tool Error Models and Error Compensation Procedure for Three Axes Vertical Machining Center Using Rigid Body Kinematics. International Journal of Machine Tools & Manufacture, 40(8), pp.1199-1213.

[11] Zeid, I. (2004). Mastering CAD/CAM. McGraw-Hill, New York, pp.848-849.

[12] Zhang, H. C. and Mei, J. (1995). Automated Tolerance Analysis for CAPP System. The International Journal of Advanced Manufacturing Technology, 10(4), 219-224.

[13] Ramesh, R., Mannan, M.A., & Poo, A.N. (2000a). Error compensation in machine tools- a review part I: geometric, cutting force induced and fixture dependent errors. International journal of machine tools and manufacture, 40(9), 1235-1256.

[14]   Ramesh, R., Mannan, M.A., & Poo, A.N. (2000b). Error compensation in machine tools- a review Part II: thermal errors. International journal of machine tools and manufacture, 40(9), 1257-1248.

[15]   S. Lou, C. Chen, and M. Li, 1999, "Surface roughness prediction technique for CNC end milling", Journal of industrial technology, 15.

[16]   X. Feng, X. Wang, 2002, "Development of empirical models for surface roughness prediction in finish turning", International Journal of Advanced Manufacturing Technology, vol. 20, pp. 348–356.

[17]   T. Ozel, Y. Karpat, 2005, "Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks, International journal of machine tools and manufacture", vol. 45, pp.467-479.

[18]   P. V. S. Suresh, P. V. Rao, and S.G. Deshmukh, 2002, "A genetic algorithmic approach for optimization of surface roughness prediction model", International journal of machine tools and manufacture, vol. 42, pp. 675-680.

[19]   S. S. Heragu, Facilities Design, Third Edition, 3rd ed. CRC Press, 2008.

[20]   Y. Koren and A. G. Ulsoy, "Reconfigurable manufacturing system having a production capacity method for ...," U.S. Patent 6349237.

[21]   F. E. Meyers and M. P. Stephens, Manufacturing Facilities Design and Material Handling, 3rd ed. Prentice Hall, 2004.

[22]   Material Handling Industry of America (MHIA) http://www.mhia.org/

[23]   Cutkosky, M.R., "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol.5, no.3, pp.269-279, Jun 1989, doi: 10.1109/70.34763. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=34763&isnumber=1436

[24]   Gordon, C.C., Churchill, T., Clauser, C.E., Bradtmiller, B., McConville, J.T., Tebbetts, I., Walker, R.A.; "1988 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics", Technical Report, 1988.

[25]   Long, E., Misra, A., Sztipanovits, J. , "Increasing productivity at Saturn," *Computer*, 31(8), pp.35-43, Aug. 1998.

[26]   Moore, J. Steven and Garg, Arun , "The Strain Index: A Proposed Method To Analyze Jobs For Risk Of Distal Upper Extremity Disorders", American Industrial Hygiene Association Journal, 56:443-458 (1995).

[27]   Ergoweb® web site at http://www.ergoweb.com/news/detail.cfm?id=583

APPENDIX A – Human Process Capability Taxonomy

Activity	Object Category	Plant Equipment, Material and Tool	Generic Assembly Tasks (Verbs)	Hand Posture	Time	Description of Task	Reach Assessment
Adjust	PPE	Belt Buckle Cover	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Ear Plugs /Protection	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Glasses	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Gloves	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Hard Hat	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Assemble	Part or Assembly		Obtain, Gain Control, Align, Insert	Five Finger Pinch Grip	move get control move put	Obtain, gain control, align and insert a part or an assembly	yes
Adjust	PPE	Protective Suit	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Ring Cover	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Sleeves	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Watch Cover	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Adjust	PPE	Work Boots	Obtain, Align	Precision Grip	move get move put	Adjust PPE	yes
Apply	Material	Label (with Backing)	Obtain, Label with Backing	Precision Grip	move get move put	Obtain, Label with Backing and apply	yes
Apply	Material	Label (without Backing)	Obtain, Label without Backing	Precision Grip	move get move put	Obtain, Label without Backing and apply	yes
Apply	Material	Tape (with Backing)	Obtain, Tape with Backing	Precision Grip	move get move put	Obtain, Tape with Backing and apply	yes
Apply	Material	Tape (without Backing)	Obtain, Tape without Backing	Precision Grip	move get move put	Obtain, Tape without Backing and apply	yes
Assemble	Tool	Drift Pin	Obtain, Align, Insert	Maximum Grip	move get move put load	Obtain Drift Pin, Align, Insert to assemble	yes
Assemble	Tool	Hook Hoist	Obtain, Align, Activate	Thumb Index Finger Pinch Grip	move get move put load	Obtain Hook Hoist, Align, Activate to assemble	yes
Assemble	Tool	Lift Assist	Obtain, Align, Activate	Thumb Index Finger Pinch Grip	move get move put load	Obtain Lift Assist, Align, Activate to assemble	yes
Assemble	Part or Assembly		Obtain, Gain Contrl, Align, Press Fit	Precision Grip	move get control move put	Obtain, Gain Contrl, Align, Press Fit a part or assembly	yes
Assemble	Part or Assembly		Obtain, Gain Control, Align, Loose Assemble	Precision Grip	move get control move put move get move put	Obtain, Gain Control, Align, Loose Assemble a part or assembly	yes
Assemble	Tool	Prop Rod	Obtain, Align, Insert	Five Finger Pinch Grip	move get move put	Obtain Prop Rod, Align, Insert to assemble	yes
Bending	Tool	Prybar	Obtain, Pry	Full Hand Power Grip	move get move put load	Obtain Prybar, Pry to assemble	yes
Bending	Tool	Hammer	Obtain, Align, Strike	Full Hand Power Grip	move get move put load	Obtain Hammer, Align, Strike to bend	yes
Bending	Tool	Mallet	Obtain, Align, Strike	Full Hand Power Grip	move get move put load	Obtain Mallet, Align, Strike to bend	yes
Bending	Part or Assembly		Obtain, Bend with Hands	Five Finger Pinch Grip	move get move put	Obtain part or assembly, Bend with Hands	yes
Bending	Tool	Prybar	Obtain, Pry	Full Hand Power Grip	move get move put load	Obtain Prybar, Pry to bend	yes
Bending	Tool	Roller Tool	Obtain, Align, Roll, Apply Force	Full Hand Power Grip	move get move put	Obtain Roller Tool, Align, Roll, Apply Force	yes
Bonding	Material	Adhesive-Glue	Obtain, Align, Smooth with Tool (repeat)	Precision Grip	move get move put	Obtain Adhesive-Glue, Align, Smooth with Tool (repeat) to bond	yes
Bonding	Tool	Brush	Obtain, Align, Brush with Tool	Full Hand Power Grip	move get move put	Obtain Brush, Align, Brush with Tool to bond	yes
Bonding	Tool	Dispensing gun	Obtain, Align, Activate	Precision Initial	move get move put	Obtain Dispensing gun, Align, Activate to bond	yes
Bonding	Part or Assembly		Obtain, Align, Push	Full Hand Power Grip	move get move put load	Obtain Part or Assembly, Align, Push to bond	yes
Bonding	Tool	Putty Knife	Obtain, Align, Slide	Full Hand Power Grip	move get move put load	Obtain Putty Knife, Align, Slide to bond	yes
Check	Computer	Computer	Visually Verify	Neutral	eye read	Visually Verify computer screen	yes
Check	Tool	Marker	Obtain, Align, Mark	Precision Grip	move get move put write	Obtain Marker, Align, Mark to check	yes
Check	Tool	Mirror	Obtain, Align, Visually Inspect	Full Hand Power Grip	move get move put read	Obtain Mirror, Align, Visually Inspect	yes
Check	Part or Assembly		Check	Neutral	eye	Check part or assembly	
Check	Tool	Snap Gage	Obtain, Align, Visually Verify	Five Finger Pinch Grip	move get move put read	Obtain Snap Guage, Align, Visually Verify	yes
Clamping	Part or Assembly		Obtain, Align, Clamp	Squeeze	move get move put load	Obtain, Align, Clamp a part or assembly	yes
Clamping	Tool	Roller Tool	Obtain, Align, Roll, Apply Force	Full Hand Power Grip	move get move put	Obtain Roller Tool, Align, Roll, Apply Force	yes
Cleaning	Tool	Brush	Obtain, Align, Brush with Tool	Full Hand Power Grip	move get move put	Obtain Brush, Align, Brush with Tool to clean	yes
Cleaning	Tool	Cleaner	Obtain, Align, Pour	Squeeze	move get move put	Obtain Cleaner, Align, Pour to clean	yes
Cleaning	Part or Assembly		Brush with Hand (repeat)	Neutral	move put move put	Brush with Hand (repeat) a part or assembly to clean	yes
Cleaning	Tool	Rag	Obtain, Align, Wipe	Fist	move get move put	Obtain Rag, Align, Wipe to clean	yes
Cleaning	Tool	Spray Bottle (Pump)	Obtain, Align, Squeeze	Squeeze	move get move put	Obtain Spary Bottle, Align, Squeeze to clean	yes
Cleaning	Tool	Squeegee	Obtain, Align, Slide	Five Finger Pinch Grip	move get move put	Obtain Squeegee, Align, Slide to clean	yes
Cleaning	Material	Water	Obtain, Align, Pour	Full Hand Power Grip	move get move put	Obtain Water, Align, Pour to clean	yes
Coating	Tool	Brush	Obtain, Align, Brush with Tool	Full Hand Power Grip	move get move put	Obtain Brush, Align, Brush with Tool to coat	yes
Coating	Part or Assembly		Obtain, Align, Dip	Precision Grip	move get move put	Obtain, Align, Dip a part or assembly	yes
Coating	Tool	Spray Gun	Obtain, Align, Activate	Precision Initial	move get move put	Obtain Spray Gun, Align, Activate to coat	yes
Cutting	Tool	Scissors	Obtain, Align, Separate, Cut	Squeeze	move get move put	Obtain Scissors, Align, Separate, Cut	yes
Cutting	Tool	Side-cutter	Obtain, Align, Separate, Apply Force	Squeeze	move get move put load	Obtain Side-Cutter, Align, Separate, Apply Force to cut	yes
Delay	Vehicle	Bike	Delay	Neutral	move wait	Wait for Bike	
Delay	Vehicle	Cart	Delay	Neutral	move wait	Wait for Cart	
Delay	Vehicle	Fork truck	Delay	Neutral	move wait	Wait for Fork truck	
Delay	Part or Assembly		Delay	Neutral	move wait	Wait for Part or Assembly	
Delay	Vehicle	Tugger	Delay	Neutral	move wait	Wait for Tugger	
Delay			Delay		move wait	Wait for Miscellaneous	
Disassemble	Tool	Inline Power Tool	Obtain, Align, Activate	Full Hand Power Grip	move get move put load	Obtain Inline Power Tool, Align, Activate to disassemble	yes
Disassemble	Tool	Pistol Grip Tool	Obtain, Align, Activate	Full Hand Power Grip	move get move put	Obtain Pistol Grip Power Tool, Align, Activate to disassemble	yes
Disassemble	Tool	Right Angle Power Tool	Obtain, Align, Activate	Full Hand Power Grip	move get move put	Obtain Right Angle Power Tool, Align, Activate to disassemble	yes
Disassemble	Tool	Tube Nut Runners	Obtain, Align, Activate	Full Hand Power Grip	move get move put	Obtain Tube Nutrunners, Align, Activate to disassemble	yes

Approved for public release; distribution unlimited.

Activity	Object Category	Plant Equipment, Material and Tool	Generic Assembly Tasks (Verbs)	Hand Posture	Time	Description of Task	Reach Assessment
Documentation	Material	Paper	Obtain, Tear	Five Finger Pinch Grip	move get move put move put	Obtain Paper, Tear for documentation	yes
Documentation	Part or Assembly	Paper	Obtain, Tape without backing, Align	Five Finger Pinch Grip	move get move put move put	Obtain, Tape without backing, Align to a part or assembly	yes
Drilling	Tool	Pistol Grip Power Tool	Obtain, Align, Activate	Precision Initial	move get move put	Obtain Pistol Grip Power Tool, Align, Activate to perform drilling	yes
Drilling	Tool	Plews Oiler	Obtain, Align, Squeeze	Squeeze	move get move put	Obtain Plews Oiler, Align, Squeeze to perform drilling	yes
Drilling	Tool	Spray Bottle (Pump)	Obtain, Align, Squeeze	Squeeze	move get move put	Obtain Spray Bottle, Align, Squeeze to perform drilling	yes
Electrical work	Part or Assembly		Obtain, Route, Connect	Five Finger Pinch Grip	move get load move put	Obtain, Route a part or assembly	yes
Electrical work	Part or Assembly		Obtain, Unroute	Precision Initial	move get load move put	Obtain, Unroute a part or assembly	yes
Electrical work	Tool	Solder Gun	Obtain, Align, Activate	Precision Grip	move get load move put	Obtain Solder Gun, Align, Activate to perform electrical work	yes
Electrical work	Material	Ties (zip or ban)	Obtain, Route, Connect	Thumb Index Pinch Grip	move get load move put	Obtain Zip Ties, Route, Connect to perform electrical work	yes
Electrical work	Part or Assembly		Verify Connection (pull test)		move get load move put	Verify Connection (pull test) to ensure electrical work is complete	yes
Fasten	Tool	Air Operated Headers Power Tool	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Air Operated Headers Power Tool, Align, Activate to fasten	yes
Fasten	Tool	Power (Manual/Automatic Reverse) Dr	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Air Power (Manual/Automatic Reverse) Driver, Align, Activate to fasten	yes
Fasten	Tool	Air-Hydraulic Headers Power Tool	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Air-Hydraulic Headers Power Tool, Align, Activate to fasten	yes
Fasten	Tool	Allen Wrench	Obtain, Align, Turn with Hand	Full Hand Power Grip	move get move put	Obtain Allen Wrench, Align, Turn with Hand to fasten	yes
Fasten	Tool	Battery Powered Drivers	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Battery Powered Drivers, Align, Activate to fasten	yes
Fasten	Tool	Bungee (Elastic Strap)	Obtain, Align, Stretch, Align	Thumb Index Pinch Grip	move get move put	Obtain Bungee, Align, Stretch, Align to fasten	yes
Fasten	Tool	CAM-Lock Wrench	Obtain, Align, Turn with Hand	Full Hand Power Grip	move get move put	Obtain CAM-Lock Wrench, Align, Turn with Hand to fasten	yes
Fasten	Tool	Channel Locks	Obtain, Align, Slide, Turn with Hand	Maximum Grip	move get move put load	Obtain Channel Locks, Align, Slide, Turn with Hand to fasten	yes
Fasten	Tool	Torque Controlled Transducerized "Sma	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Critical Torque Controlled Transducerized "Smart" Tools, Align, Activa	yes
Fasten	Tool	Inline Power Tool	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Inline Power Tool, Align, Activate to fasten	yes
Fasten	Tool	Multiple Spindle Air Drivers	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Multiple Spindle Air Drivers, Align, Activate to fasten	yes
Fasten	Tool	ple Spindle Electrical Transducerized D	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Multiple Spindle Electrical Transducerized Drivers, Align, Activate to f	yes
Fasten	Tool	Oil Pulse Driver	Obtain, Align, Activate	Full Hand Power Grip	move get move put	Obtain Oil Pulse Driver, Align, Activate to fasten	yes
Fasten	Part or Assembly	One-hander Pliers	Obtain, Separate, Align, Clamp	Full Hand Power Grip	move get move put	Obtain One-hander Pliers, Separate, Align, Clamp to fasten	yes
Fasten	Part or Assembly		Obtain, Align, Push	Neutral	move get move put move put	Obtain, Align, Push on part or assembly to fasten	yes
Fasten	Tool	Pistol Grip Power Tool	Obtain, Clip	Thumb Index Five Finger Pinch Grip	move get move put	Obtain, Clip a part or assembly to fasten	yes
Fasten	Tool	Pliers	Obtain, Align, Activate	Precision Initial	move get move put	Obtain Pistol Grip Power Tool, Align, Activate to fasten	yes
Fasten	Tool	Ratchet (box end) Wrench	Obtain, Separate, Align, Clamp	Full Hand Power Grip	move get move put	Obtain Pliers, Separate, Align, Clamp to fasten	yes
Fasten	Tool	Right Angle Power Tool	Obtain, Align, Turn with Hand	Full Hand Power Grip	move get move put move put	Obtain Ratchet Wrench, Align, Turn with Hand to fasten	yes
Fasten	Tool	Rivet Gun	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Right Angle Power Tool, Align, Activate to fasten	yes
Fasten	Tool	Screwdriver	Obtain, Align, Turn with Hand	Full Hand Power Grip	move get move put	Obtain Rivet Gun, Align, Activate to fasten	yes
Fasten	Tool	e Spindle Air Drivers (Manual/Auto rev	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Screwdriver, Align, Turn with Hand to fasten	yes
Fasten	Tool	ingle Spindle Air Transducerized Driver	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Single Spindle Air Driver, Align, Activate to fasten	yes
Fasten	Tool	le Spindle Electrical Transducerized Dri	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Single Spindle Air Transducerized Drivers, Align, Activate to fasten	yes
Fasten	Tool	Socket Handle	Obtain, Align, Turn with Hand	Full Hand Power Grip	move get move put	Obtain Single Spindle Electrical Transducerized Drivers, Align, Activate to fas	yes
Fasten	Tool	Socket Wrench (Tee-handle)	Obtain, Align, Turn with Hand	Full Hand Power Grip	move get move put	Obtain Socket Handle, Align, Turn with Hand to fasten	yes
Fasten	Tool	Speed (open end) Wrench	Obtain, Align, Turn with Hand	Full Hand Power Grip	move get move put	Obtain Socket Wrench, Align, Turn with Hand to fasten	yes
Fasten	Tool	Staple Gun	Obtain, Align, Palm Press	Squeeze	move get move put	Obtain Speed Wrench, Align, Turn with Hand to fasten	yes
Fasten	Tool	Tube Nut Runners	Obtain, Align, Activate	Maximum Grip	move get move put	Obtain Staple Gun, Palm Press to fasten	yes
Fasten	Tool	Yankee (Push) Screwdriver	Obtain, Align, Activate	Full Hand Power Grip	move get move put	Obtain Tube Nut Runner, Align, Activate to fasten	yes
Fill	Tool	Fluid Fill Nozzle	Obtain, Align, Activate	Squeeze	move get move put	Obtain Yankee Screwdriver, Align, Activate to fasten	yes
Form	Part or Assembly		Stretch	Five Finger Pinch Grip	move get move put	Obtain Fluid Fill Nozzle, Align, Activate to fill	yes
Form	Tool	Roller Tool	Obtain, Align, Roll, Apply Force	Full Hand Power Grip	move get move put load	Stretch part or assembly to form	yes
Grind	Tool	Brush	Obtain, Align, Brush with Tool	Full Hand Power Grip	move get move put	Obtain Roller Tool, Align, Roll, Apply Force to form	yes
Inspect	Tool	Marker	Obtain, Align, Mark	Precision Grip	move get write	Obtain Brush, Align, Brush with Tool to grind	yes
Inspect	Tool	Mirror	Obtain, Align, Visually Inspect	Full Hand Power Grip	move get eye read	Obtain Marker, Align, Mark to inspect	yes
Inspect	Tool	Snap Gage	Obtain, Align, Visually Verify	Five Finger Pinch Grip	move get eye read	Obtain Mirror, Align, Visually Inspect	yes
Install	Material	Adhesive-Glue	Obtain, Align, Smooth with Tool (repeat)	Precision Grip	move get move put	Obtain Snap Gauge, Align, Visually Verify	yes
Install	Tool	Hook Hoist	Obtain, Align, Activate	Thumb Index Five Finger Pinch Grip	move get move put	Obtain Adhesive-Glue, Align, Smooth with Tool (repeat) to install	yes
Install	Tool	Lift Assist	Obtain, Align, Activate	Squeeze	move get move put	Obtain Hook Hoist, Align, Activate to install	yes
Install	Tool	Plews Oiler	Obtain, Align, Squeeze	Full Hand Power Grip	move get move put	Obtain Lift Assist, Align, Activate to install	yes
Install	Tool	Prop Rod	Obtain, Align, Insert	Five Finger Pinch Grip	move get move put	Obtain Plews Oiler, Align, Squeeze to install	yes
Install	Tool	Spray Bottle (Pump)	Obtain, Align, Squeeze	Squeeze	move get move put	Obtain Prop Rod, Align, Insert to install	yes
Machine	Tool	Plews Oiler	Obtain, Align, Squeeze	Squeeze	move get move put	Obtain Spray Bottle, Align, Squeeze to install	yes
Machine	Tool	Punch	Obtain, Align, Apply Force	Full Hand Power Grip	move get move put	Obtain Plews Oiler, Align, Squeeze to machine	yes
Mark	Tool	Marker	Obtain, Align, Mark	Precision Grip	move get write	Obtain Punch, Align, Apply Force to machine	yes
Measure	Part or Assembly		Visually Verify	Neutral	move get eye read	Obtain Marker, Align, Mark	yes
Measure	Tool	Snap Gage	Obtain, Align, Visually Verify	Five Finger Pinch Grip	move get eye read	Visually Verify measurement of part or assembly	yes
Plumbing	Tool	Solder Gun	Obtain, Align, Activate	Precision Initial	move get move put	Obtain Snap Guage, Align, Visually Verify	yes
Prep Work	Tool	Fender Protector Pad (right and left)	Obtain, Align, Place	Five Finger Pinch Grip	move get move put	Obtain Solder Gun, Align, Activate to perform plumbing work	yes
						Obtain Fender Protector Pad, Align, Place to prep vehicle	yes

Activity	Object Category	Plant Equipment, Material and Tool	Generic Assembly Tasks (Verbs)	Hand Posture	Time	Description of Task	Reach Assessment
Prep Work	Equipment	Lift Table	Actuate Foot Pedal	Neutral	foot	Actuate Foot Pedal to manipulate lift table	yes
Prep Work	Equipment	Container	Obtain, Align, Place	Five Finger Pinch Grip	get move put	Obtain Container, Align, Place to prep workcell	yes
Prep Work	Equipment	Dunnage	Obtain, Align, Place	Five Finger Pinch Grip	get move put	Obtain Dunnage, Align, Place to prep workcell	yes
Prep Work	Equipment	Machine	Obtain, Align, Place	Five Finger Pinch Grip	get move put	Obtain, Align, Place item to/from machine from/to workcell	yes
Prep Work	Equipment	Pallet	Obtain, Align, Place	Five Finger Pinch Grip	get move put	Obtain, Align, Place item to/from pallet from/to workcell	yes
Prep Work	Equipment	Tote / Bin for Kitting	Obtain, Align, Place	Five Finger Pinch Grip	get move put	Obtain, Align, Place item to/from Tote from/to workcell	yes
Prep Work	Material	Lubricants	Obtain, Align, Pour	Squeeze	get move put	Obtain Lubricant, Align, Pour to prep work	yes
Prep Work	Equipment	Container	Obtain, Aside	Five Finger Pinch Grip	get move move put	Obtain Container, Aside to prep workcell	yes
Prep Work	Equipment	Dunnage	Obtain, Aside	Five Finger Pinch Grip	get move move put	Obtain Dunnage, Aside to prep workcell	yes
Prep Work	Equipment	Pallet	Obtain, Aside	Five Finger Pinch Grip	get move move put	Obtain, Aside Items to/from pallet from/to workcell	yes
Prep Work	Equipment	Tote / Bin for Kitting	Obtain, Aside	Five Finger Pinch Grip	get move move put	Obtain, Aside Items to/from Tote from/to workcell	yes
Prep Work	Material	Sand Paper	Obtain, Align, Sand	Thumb Press	get move move put	Obtain Sand Paper, Align, Sand	yes
Prep Work	Tool	Spray Bottle (Pump)	Obtain, Align, Squeeze	Squeeze	get move move put	Obtain Spray Bottle, Align, Squeeze to prep work	yes
Print	Computer	Computer	Keystroke	Precision Initial	keystroke	Keystroke button to print	
Print	Material	Paper	Obtain, Tear	Five Finger Pinch Grip	get move put	Obtain Paper, Tear printout (manifest)	yes
Punch Hole	Tool	Punch	Obtain, Align, Apply Force	Full Hand Power Grip	get move put	Obtain Punch, Align, Apply Force to punch hole	yes
Put On	PPE	Belt Buckle Cover	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Ear Plugs /Protection	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Glasses	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Gloves	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Hard Hat	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Protective Suit	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Ring Cover	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Sleeves	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Watch Cover	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Put On	PPE	Work Boots	Obtain, Align	Precision Grip	get move put	Obtain PPE, Align to put on	yes
Remove	PPE	Belt Buckle Cover	Obtain, Remove	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	PPE	Ear Plugs /Protection	Obtain, Remove	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	PPE	Glasses	Obtain, Align	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	PPE	Gloves	Obtain, Remove	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	PPE	Hard Hat	Obtain, Remove	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	Part or Assembly		Unfasten	Five Finger Pinch Grip	get move put	Unfasten part or assembly	yes
Remove	Part or Assembly		Unhook	Five Finger Pinch Grip	load put	Unhook part or assembly	yes
Remove	Part or Assembly		Unclip	Five Finger Pinch Grip	load put	Unclip part or assembly	yes
Remove	PPE	Protective Suit	Obtain, Remove	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	Tool	Prybar	Obtain, Pry	Full Hand Power Grip	get move put	Obtain Prybar, Pry	yes
Remove	PPE	Ring Cover	Obtain, Remove	Pinch	get move put	Obtain PPE, Remove	yes
Remove	Material	Shipping Caps	Obtain, Remove, Aside	Precision Grip	get move move put	Obtain Shipping Caps, Remove, Aside	yes
Remove	PPE	Sleeves	Obtain, Remove	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	Material	Tape	Obtain, Tape without Backing	Precision Grip	get move put	Obtain, Tape without Backing and remove	yes
Remove	PPE	Watch Cover	Obtain, Remove	Precision Grip	get move put	Obtain PPE, Remove	yes
Remove	PPE	Work Boots	Obtain, Remove	Five Finger Pinch Grip	get move put	Obtain PPE, Remove	yes
Return	Equipment	Conveyor	Obtain, Place	Five Finger Pinch Grip	get move move put	Obtain, Place to return item to conveyor	yes
Return	Equipment	Machine	Obtain, Place	Full Hand Power Grip	get move move put	Obtain, Place to return item to machine	yes
Return	Part or Assembly		Disengage	Precision Initial	move load put	Disengage part or assembly to return item (i.e. lift assist)	yes
Return	Part or Assembly		Dispose (single or multiple objects)		move put	Dispose (single or multiple objects)	yes
Ride	Vehicle	Bike	Process Dependent	Steering Grip	process	Ride Bike	
Ride	Vehicle	Cart	Process Dependent	Steering Grip	process	Ride Cart	
Ride	Vehicle	Fork truck	Process Dependent	Steering Grip	process	Ride Fork truck	
Ride	Vehicle	Tugger	Process Dependent	Steering Grip	process	Ride Tugger	
Rig	Equipment	Crane	Obtain, Align, Place	Five Finger Pinch Grip	get move put	Obtain Crane (rigging), Align, Place to rig	yes
Rig	Part or Assembly		Obtain, Align, Hang	Full Hand Power Grip	get move put	Obtain Part or Assembly, Align, Hang	yes
Rig	Part or Assembly		Obtain, Align, Hook	Full Hand Power Grip	get move put	Obtain Part or Assembly, Align, Hook	yes
Rig	Equipment	Crane	Obtain, Remove	Five Finger Pinch Grip	get move put	Obtain Crane (rigging), Remove	yes
Sand	Tool	Brush	Obtain, Align, Brush with Tool	Full Hand Power Grip	get move move put	Obtain Brush, Align, Brush with Tool to sand	yes
Scrape	Tool	Putty Knife	Obtain, Align, Slide	Full Hand Power Grip	get move move put	Obtain Putty Knife, Align, Slide to scrape	yes
Seal	Tool	Dispensing gun	Obtain, Align, Activate	Precision Initial	get move move put	Obtain Dispensing Gun, Align, Activate to seal	yes
Seal	Tool	Seal Tool	Obtain, Align, Slide	Full Hand Power Grip	get move move put	Obtain Seal Tool, Align, Slide to seal	yes
Search	Computer	Computer	Visually Inspect		eye read	Visually Inspect computer	
Search	Tool	Mirror	Obtain, Align, Visually Inspect	Full Hand Power Grip	get move put	Obtain Mirror, Align, Visually Inspect	yes
Set Up	Equipment	Lift Table	Actuate Foot Pedal	Neutral	foot	Actuate Foot Pedal to manipulate lift table	yes

Activity	Object Category	Plant Equipment, Material and Tool	Generic Assembly Tasks (Verbs)	Hand Posture	Time (Motions)	Description of Task	Reach Assessment
Set Up	Equipment	Container	Obtain, Align, Place	Five Finger/Pinch Grip	move get move put	Obtain Container, Align, Place to set up workcell	yes
Set Up	Equipment	Dunnage	Obtain, Align, Place	Five Finger/Pinch Grip	move get move put	Obtain Dunnage, Align, Place to set up workcell	yes
Set Up	Equipment	Machine	Obtain, Align, Place	Five Finger/Pinch Grip	move get move put	Obtain, Align, Place item to/from machine from/to workcell	yes
Set Up	Equipment	Pallet	Obtain, Align, Place	Five Finger/Pinch Grip	move get move put	Obtain, Align, Place item to/from pallet from/to workcell	yes
Set Up	Equipment	Tote/Bin for Kitting	Obtain, Align, Place	Five Finger/Pinch Grip	move get move put	Obtain, Align, Place item to/from Tote from/to workcell	yes
Set Up	Part or Assembly		Untangle (repeat as needed)	Five Finger/Pinch Grip	move get move put	Untangle (repeat as needed) part or assembly to set up work	yes
Set Up	Tool	Plews Oiler	Obtain, Align, Squeeze	Squeeze	move get move put	Obtain Plews Oiler, Align, Squeeze to set up work	yes
Set Up	Tool	Prop Rod	Obtain, Align, Insert	Full Hand Power Grip	move get move put	Obtain Prop Rod, Align, Insert to set up work	yes
Set Up	Tool	Prybar	Obtain, Pry	Five Finger/Pinch Grip	move get move put	Obtain Prybar, Pry to set up work	yes
Set Up	Equipment	Container	Obtain, Aside	Full Hand Power Grip	move get move put	Obtain Container, Aside to set up workcell	yes
Set Up	Equipment	Dunnage	Obtain, Aside	Five Finger/Pinch Grip	move get move put	Obtain Dunnage, Aside to set up workcell	yes
Set Up	Equipment	Pallet	Obtain, Aside	Five Finger/Pinch Grip	move get move put	Obtain, Aside items to/from pallet from/to workcell	yes
Set Up	Equipment	Tote/Bin for Kitting	Obtain, Aside	Five Finger/Pinch Grip	move get move put	Obtain, Aside items to/from Tote from/to workcell	yes
Shim	Part or Assembly		Obtain, Align, Push	Five Finger/Pinch Grip	move get move put load	Obtain, Align, Push on part or assembly to shim	yes
Shim	Tool	Prybar	Obtain, Pry	Full Hand Power Grip	move get move put load	Obtain Prybar, Pry to shim	yes
Strip	Tool	Putty Knife	Obtain, Align, Slide	Full Hand Power Grip	move get move put load	Obtain Putty Knife, Align, Slide to Strip	yes
Communicate	Human Speech	Crane	Communicate	Neutral	vocalize	Communicate to maneuver crane with assistance	
Tear	Material	Paper	Obtain, Align, Tear	Precision Grip	move get move put	Obtain Paper, Align, Tear	yes
Tear	Part or Assembly		Obtain, Align, Tear	Precision Grip	move get move put	Obtain, Align, Tear part or assembly	yes
Tool Change	Equipment	Machine	Obtain, Align, Place	Five Finger/Pinch Grip	move get move put	Obtain, Align, Place item to be changed	yes
Travel	Vehicle	Bike	Travel	Steering Grip	process	Travel using Bike	
Travel	Vehicle	Cart	Travel	Steering Grip	process	Travel using Cart	
Travel	Vehicle	Fork truck	Travel	Steering Grip	process	Travel using Fork truck	
Travel	Vehicle	Tugger	Travel	Steering Grip	process	Travel using Tugger	
Travel	Vehicle		Travel		process	Travel Miscellaneous	
Wait	Vehicle	Bike	Wait	Neutral	wait	Wait for Bike	
Wait	Vehicle	Cart	Wait	Neutral	wait	Wait for Cart	
Wait	Vehicle	Fork truck	Wait	Neutral	wait	Wait for Fork truck	
Wait	Vehicle	Tugger	Wait	Neutral	wait	Wait for Tugger	
Walk	Human Motion		Travel	Neutral	walk	Human walking	
General			Arise	Neutral	bend/arise body	Arise	
General			Bend Body	Neutral	move put	Bend Body	
General			Climb (repeat)	Neutral	climb	Climb (repeat)	
General			Crawl (repeat)	Neutral	crawl	Crawl (repeat)	
General			Crumble with hand	Five Finger/Pinch Grip	move put	Crumble with hand	
General			Grasp with hand		move put	Grasp with hand	yes
General			Lay on Back	Neutral	lay on back	Lay on Back	yes
General			Lean on Hand, Elbow, Thigh, Pelvis	Neutral	move put	Lean on Hand, Elbow, Thigh, Pelvis	yes
General			Loose Grasp		move get	Loose Grasp	yes
General			Loosen (repeat)		move move put	Loosen (repeat)	yes
General			Pan	Neutral	move get move put	Pan	yes
General			Pinch Grasp		move get	Pinch Grasp	yes
General			Pound with Fist	Fist	move load	Pound with Fist	yes
General			Pull		move put load	Pull	yes
General			Pull Self Up		move put load	Pull Self Up	yes
General			Pump (repeat)		move put	Pump (repeat)	yes
General			Release		move put	Release	yes
General			Retain		move put	Retain	yes
General			Scoop		move get move put	Scoop	yes
General			Scrape with finger		move get move put	Scrape with finger	yes
General			Secure		move put move put	Secure	yes
General			Sit		move sit to/from stand	Sit	yes
General			Snap Fit		move put load	Snap Fit	yes
General			Staple		move get move put load	Staple	yes
General			Store		move put	Store	yes
General			Stow		move put	Stow	yes
General			Tighten with Hand (repeat)		move get move put	Tighten with Hand (repeat)	yes
General			Turn Body	Neutral	move put	Turn Body	yes
General			Twist Body	Neutral	move put	Twist Body	yes
General			Unstaple		move put load	Unstaple	yes

129

APPENDIX B – Boeing/GM Human Operations Schema

ACTIVITY	DESCRIPTION	DETAIL
PPE	Adjust	
PPE	Off	
PPE	On	
Assemble	One Time	
Assemble	Recurring	
Bending	Ducts	
Bending	Panels	
Bending	Tubes	
Bonding	Application	Glue
Bonding	Application	Tape
Bonding	Electrical	
Bonding	Prep	
Break down	Disconnect	
Break down	Pick-n-Place	
Break down	Position & Align	
Break down	Tools	
Check Work		
Check/Inspection	Mechanic	Visual check/Ensure
Clamping		
Clean up	5S	
Clean up	HazMat/Disposal	
Clean up	Vacuum	
Clean up	Waste Disposal	
Cleaning	Chemical	
Cleaning	Chemical	Manual
Cleaning	Chemical	Mechanized
Cleaning	HazMat/Disposal	
Cleaning	Nonchemical	
Cleaning	Polishing	
Climbing	Loaded	
Climbing	Unloaded	
Coating	Adhesive	
Coating	Anticorrosive	
Coating	Lubricant	
Coating	Paint	
Coating	Paint Primer	
Coating	Paint Top Coat	
Computer	Call for Final	

Computer	Call for IPIR	
Computer	Clock in	
Computer	Clock out	
Computer	Print Doc	
Computer	Record Data	
Computer	Request Sealer	
Computer	Review Doc	
Constraint		
Crawling	Loaded	
Crawling	Unloaded	
Cutting	Manual	
Cutting	Mechanized	
Cutting		
Delay		
Disassemble	One Time	
Disassemble	Recurring	
Documentation		
Drilling	Coldwork	
Drilling	Counterbore	
Drilling	Countersink	
Drilling	Debur	
Drilling	Line Bore	
Drilling	Milling	
Drilling	Pilot	
Drilling	Ream	
Drilling	Regular	
EdgeWrap	ApplyTape	
EdgeWrap	Smooth	
EdgeWrap	Trim	
Electrical	Coil&Stow	
Electrical	Crimp	
Electrical	Cut	Wire
Electrical	Cut	
Electrical	Insert	Pin
Electrical	Insert	
Electrical	Install	Bag
Electrical	Install	Bracket
Electrical	Install	Clamp/Post
Electrical	Install	Cntrl. Comp.
Electrical	Install	Coax
Electrical	Install	Connector

Electrical	Install	Heat Shrink
Electrical	Install	ID Tag
Electrical	Install	Sleeve/Shell
Electrical	Install	Spacer
Electrical	Install	Tape
Electrical	Lockstitch	
Electrical	Mate	
Electrical	Other	
Electrical	Remove	Connector
Electrical	Remove	Diaelectric
Electrical	Remove	Foil
Electrical	Remove	Labels
Electrical	Remove	Package
Electrical	Remove	Pckg
Electrical	Remove	Pin
Electrical	Remove	Shield
Electrical	Route	
Electrical	Solder	
Electrical	Sort/Adjust	
Electrical	Splice	
Electrical	Strip	
Electrical	Tie/Strap	
Electrical	Uncoil	
Electrical	Unroute	
Fastening	(Wet) Pick-n-Place	
Fastening	Align/Pick-n-Place	Feed
Fastening	Align/Pick-n-Place	Position
Fastening	Align/Pick-n-Place	Slide
Fastening	Clip	
Fastening	Lockwire/Cotterpin	
Fastening	Obtain/Pick-n-Place	Bring/Move
Fastening	Obtain/Pick-n-Place	Collect
Fastening	Obtain/Pick-n-Place	Hold
Fastening	Obtain/Pick-n-Place	Lower
Fastening	Pick-n-Place	
Fastening	Remove/Loosen	
Fastening	Screw	
Fastening	Seat	
Fastening	Stapling	
Fastening	Stitching	
Fastening	Tighten	Tighten

Fastening	TorqPutty	
Fastening	Torque	Hand start
Fastening	Torque	Torque screw
Fastening	Turn	
Forming	Bead	
Forming	Expanding	
Forming	Flaring	
Forming	Manual	
Forming	Mechanized	
Forming	Pullout	
Forming	Vail	
Functional Test	Calibrate	
Functional Test	Hookup equip	
Functional Test	Inspect/Shake-Out Config	
Functional Test	Load Software	
Functional Test	Post-Test Restore	
Functional Test	PreTest Removal	
Functional Test	Run/Cycle Procedure	
Functional Test	TroubleShoot	
Functional Test	Verify Pre-Test Config	
Get	Consumables	
Get	Kit	
Get	Parts	
Get	Standards	
Get	Tools	
Grind/Sand		
Inset	Pick-n-Place	
Inset	Potting	
Inset	Scrap	
Inset	Seat	
Inset	Smooth	
Inspection	Customer QA	
Inspection	Customer Rep	
Inspection	QA	
Inspection		Locate/Select
Installing	Clamp	
Installing	Grommet	
Installing	Part Pick-Up&Place	Install
Installing	Part Remove&Place	
Installing	Activate/Start	
Installing	Depress	

Installing	Engage	
Installing	Hang	
Installing	Insert	
Installing	Join	
Installing	Press fit	
Installing	Pull	
Installing	Push	
Installing	Retain	
Installing	Snap fit	
Installing	Permanent Tape	
Installing	Position/Align	
Installing	Swage	
Layup	By Hand	
Layup	By Machine	
Layup	Raise / Lower Compactor	
Layup	Rollup / Out Material	
Layup	Rub Down Course	
Machine	Idle	
Machine	Run	
Machine	Set Up	
Measuring	Back Mark	
Measuring	Continuity	
Measuring	Depth	
Measuring	Gap	
Measuring	Location	
Measuring	Size	
Meeting		
Meter	Down Load	
Micro Break		
MOP	Load/Unload	
MOP	Run	
Open/Close		
Other		
Paper	Record Data	
Paper	Review Doc	
Paper	Stamp/Sign Off	
Part Mark		
Personal	Allowance	
Personal	Break Predetermined	
Personal	Lunchtime	
Plumbing	Pick-Up Part	

Plumbing	Position/Align	
Plumbing	Remove Part	
Plumbing	Tighten/Torque	
Prep Work		
Press	Apply Release Agent	
Press	Mold Die Change	
Press	Mold Die HeatUp/CoolDown	
Press	Run	
Press	Set Up	
Print	Color Check	
Print	Digital	
Print	Ink Mix	
Print	Printline	
Punch Hole		
Removing	Part Remove&Place	
Removing	Position/Align	
Removing	Standard Remove&Place	
Return	Consumables	
Return	Kit	
Return	Parts	
Return	Standards	
Return	Tools	
Riding		
Riding Lift		
Rigging	Connect	
Rigging	Disconnect/Remove	
Rigging	Inspect/Shake-Out Config	
Rigging	Lockout/Tagout	
Rigging	Part Pick-Up&Place	
Rigging	Position/Align	
Rigging	Post-Rig Restore	
Rigging	Route	
Rigging	Run/Cycle Procedure	
Rigging	Tighten/Torque	
Rigging	TroubleShoot	
Rigging	Verify Pre-Rig Config	
Rivets	Buck/Drive	
Rivets	Drill Out Rivet	
Rivets	Microshave	
Rivets	Pick-n-Place	
Rivets	Pneumatic Installer	

Rivets	Seat	
Scraping/Stripping		
Sealing	Apply	
Sealing	Smooth	
Search	Consumables	
Search	Kit	
Search	Parts	
Search	Standards	
Search	Tools	
Set Up	Connect	
Set Up	Layout	
Set Up	Pckg Removal	
Set Up	Pick-n-Place	
Set Up	Position & Align	
Set Up	Tools	
Shimming	Anticorrosive	
Shimming	Cut/Shear/Grind	
Shimming	Fit/Check	
Shimming	Measure/Mark	
Shimming	Peel	
Talking	Job Related	
Talking	Not Job Related	
Tape Apply Remove		
Tear	Rip	
Thermal	Aging	
Thermal	Annealing	
Thermal	Baking	
Thermal	Drying	
Thermal	Freezing	
Thermal	Heat Treat	
Tool Change		
Travel		
Vacuum Form	Load/Unload	
Vacuum Form	Oven	
Vacuum Form	RSE (RightSizeEquipment)	
Waiting	Customer	
Waiting	Customer Rep	
Waiting	For Assist	
Waiting	For Cure Time	
Waiting	In Queue	
Waiting	On Machinery	

Waiting	Other	
Waiting	QA	
Walking	Loaded	
Walking	Loaded	
Walking	Unloaded	
Welding	Manual	
Welding	Mechanized	

APPENDIX C – Metamodels for Graphical Languages

Formally, a domain-specific modeling language is a five-tuple of concrete syntax (C), abstract syntax (A), semantic domain (S) and semantic and syntactic mappings (M_s , and M_c):

$$L = \langle C, A, S, M_s, M_c \rangle$$

The C concrete syntax defines the specific notation used to express models, which may be graphical, textual or mixed. The A abstract syntax defines the concepts, relationships, and integrity constraints available in the language. Thus, the abstract syntax determines all the (syntactically) correct "sentences" (i.e. models) that can be built. (It is important to note that the abstract syntax includes semantic elements as well. The integrity constraints, which define well-formedness rules for the models, are frequently called "static semantics".) The S semantic domain is usually defined by means of some mathematical formalism in terms of which the meaning of the models is explained. The $M_c : A \rightarrow C$ mapping assigns syntactic constructs (graphical, textual or both) to the elements of the abstract syntax. The $M_s : A \rightarrow S$ semantic mapping relates syntactic concepts to those of the semantic domain. The definition of the (DSM) language proceeds by constructing metamodels of the language (to cover A and C), and by constructing a metamodel for the semantics (to cover M_c and M_s).

The Generic Modeling Environment (GME)[†] uses an UML-based approach to define modeling languages. The underlying assumption is that graphical modeling languages have "sentences" formed from objects, i.e. a sentence is a network of objects. A UML class diagram can capture classes, their attributes, and their relationships: inheritance, containment, and general associations. A programmer can instantiate those classes, specify instance attributes, and establish links among objects that correspond to associations in the class diagram.

Unfortunately, pure UML class diagrams are not well suited for the metaprogramming of modeling environments. The reason is that environments tend to support some core modeling concepts (e.g. containers, ported objects, atomic objects, etc.), which are not UML concepts, yet metamodels should contain hints how UML class diagrams should be interpreted in terms of those concepts. A convenient solution to this problem is to use stereotypes, which mark classes as belonging to a specific category that is meaningful (and has a specific semantics in) the modeling environment. In the metaprogrammable Generic Modeling Environment this approach has been followed. Figure C-1 below illustrates how a UML class diagram can be embellished to define a meta-model for GME. The drawing also summarizes the core model organization concepts supported by GME. The diagram, read as a pure UML diagram, has the following classes: aFolder: an untyped container of objects, aModel: a typed container with model semantics, anAtom and anotherAtom: simple objects, aConnection: an association class relating the classes anAtom and anotherAtom, anotherModel: a container for anotherAtoms and anotherModels, aSet: yet another container containing aSetElement, and aReference: associates with ("points to") anotherAtoms. The stereotypes map these classes into environment-specific

---

[†] Ledeczi, A.; Bakay, A.; Maroti, M.; Volgyesi, P.; Nordstrom, G.; Sprinkle, J.; Karsai, G.: Composing domain-specific design environments, IEEE Computer, Nov. 2001, Page(s): 44 –51.

Approved for public release; distribution unlimited.

modeling concepts. GME supports <<Model>>-s, which are composite objects with ports containing other objects (including other <<Model>>-s), <<Atom>>-s are primitive objects that have their own graphical icons, <<Set>>-s are special containers that contain objects within the same parent <<Model>> that also contains the set, <<References>> are alias objects which point to (non-local) objects in the object hierarchy, and <<Connection>>-s are association objects relating two (or more) iconic objects. All objects except the <<Connection>>-s are iconic. <<Model>>-s can have ports on their icons, and <<Connection>>-s are visualized as lines. It is not shown on the drawing, but many stereotypes have a corresponding "proxy" stereotype, which is semantically equivalent to the base stereotype.
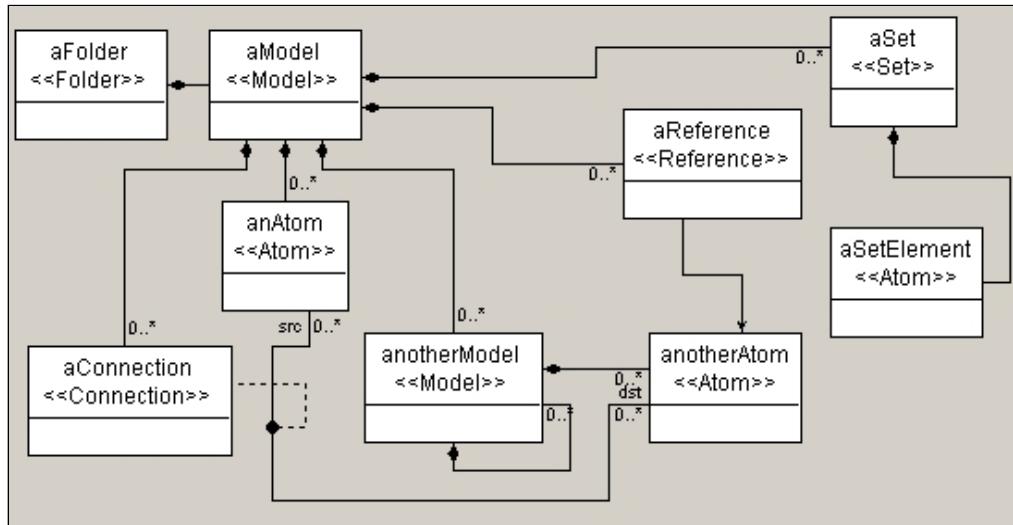

Figure C-1.   Example Meta-Model

Approved for public release; distribution unlimited.

APPENDIX D – Surface Roughness Evaluation Literature Review

Lou ([15]) proposed multiple regression analysis for surface roughness prediction in end milling operations. The surface roughness could be predicted by applying spindle speed, feed rate, depth of cut, and their interactions in the multiple regression model. The proposed model is a three way interaction equation:

$$R_a = \alpha_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{1i} X_{2i} + \beta_5 X_{1i} X_{3i} + \beta_6 X_{2i} X_{3i} + \beta_7 X_{1i} X_{2i} X_{3i} \quad (2)$$

where, R_a is the surface roughness (μm); X_{1i} is the spindle speed; X_{2i} is the feed rate; X_{3i} is the depth of cut; $\alpha_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7$, are parameters need to be optimized.

Feng ([16]) conducted testing and used regression analysis to develop a complete empirical model of surface roughness for traditional turning. The model considers the following working parameters: workpiece hardness (material); feed rate; cutting tool point angle; depth of cut; spindle speed; and cutting time. An exponential model was suggested for surface roughness including workpiece hardness (*H*), cutting tool point angle (*A*), cutting speed (*V*), feed (*f*), depth of cut (*d*), and cutting time (*T*) to account for tool life, hence surface roughness:

$$R_a = c_0 H^{c_1} A^{c_2} V^{c_3} f^{c_4} d^{c_5} T^{c_6} \quad (3)$$

where c_0 is a constant, and $c_1, c_2, c_3, c_4, c_5, c_6$ are parameters need to be obtained.

Suresh ([18]) utilized genetic algorithm for optimization of surface roughness prediction model. Response surface methodology (RSM) was used in the study for development of a surface roughness prediction model for machining mild steel. Surface roughness was considered to be affected by cutting speed (*V*), feed (*f*), depth of cut (*d*), and tool nose radius (*r*). The relationship between surface roughness and those variables is modeled as:

$$R_a = c_0 V^{c_1} f^{c_2} d^{c_3} r^{c_4} \quad (4)$$

where c_0 is a constant, and c_1, c_2, c_3, c_4 are parameters need to be obtained.

Ozel ([17]) developed a modified exponential model for surface roughness. In the model, surface roughness is function of work material hardness, CBN content in tool material, edge radius of the CBN cutting tool, cutting speed, feed and cutting time:

$$R_a = c_0 H^{c_1} C^{c_2} E^{c_3} V^{c_4} f^{c_5} L^{c_6} \quad (5)$$

Where R_a is surface roughness (μm), *H* is work material hardness in Rockwell-C scare, *E* is edge radius of the CBN tool (μm), *C* is CBN content in percentage volume, f is feed (mm/rev), *V* is cutting speed (m/min), L is cutting length in axial direction (mm). c_0 is a constant, and $c_1, c_2, c_3, c_4, c_5, c_6$ are parameters need to be obtained
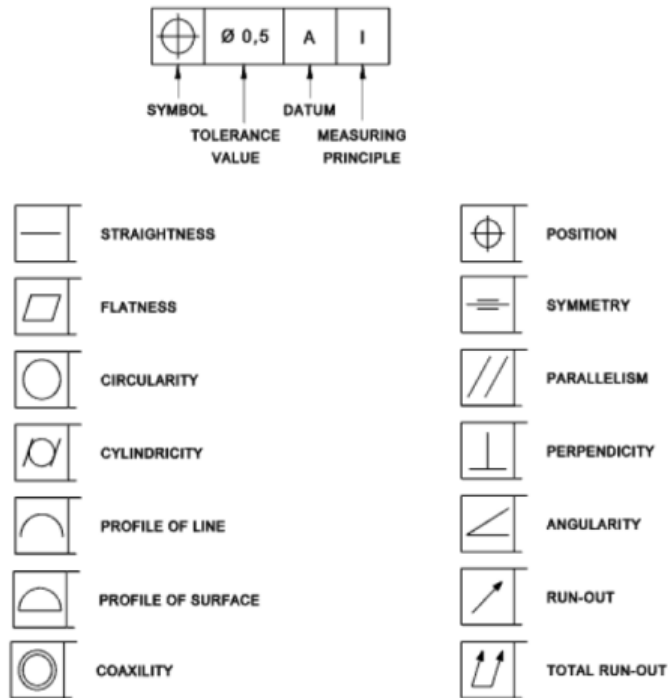
Approved for public release; distribution unlimited.

Figure D-1.   Geometric tolerancing and dimensioning schemes and types.

Table 1. Relation of Machining processes to IT Tolerance Grades

Tolerance grade	Manufacturing process and application	Machine required
IT 01, IT0 IT1 too IT5	Super finishing process, such as lapping, diamond boring etc. Use: Gauges	Super finishing machines
IT6	Grinding	Grinding machine
IT7	Precision turning, broaching, honing	Boring machine, honing machine
IT8	Turning, boring, and reaming	Lathes, capstan, honing machine
IT9	Boring	Boring machine
IT10	Milling, slotting, planning, rolling and extrusion	Milling machine, slotting machine, planning machine and extruders

Approved for public release; distribution unlimited.

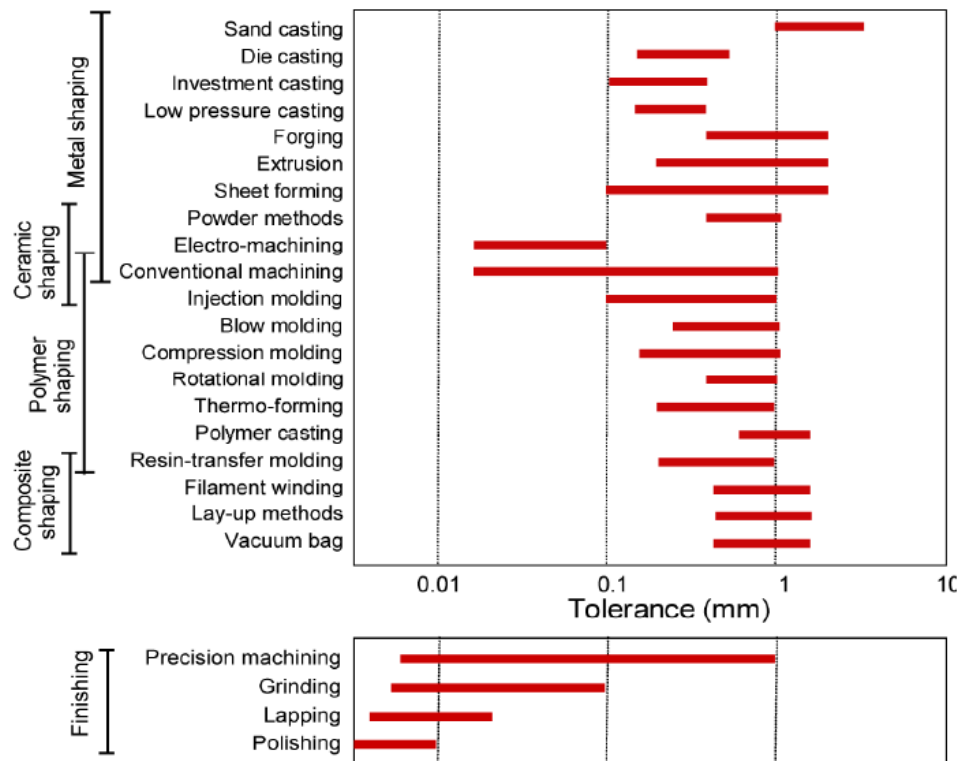| IT11	Drilling, rough turning	Drilling machine, lathes
IT12, IT13, IT14	Metal forming processes	Presses
IT15	Die casting, stamping	Die casting machine, hammer machine
IT16	Sand casting	

Figure D-2. **Tolerance ranges for different machining processes**
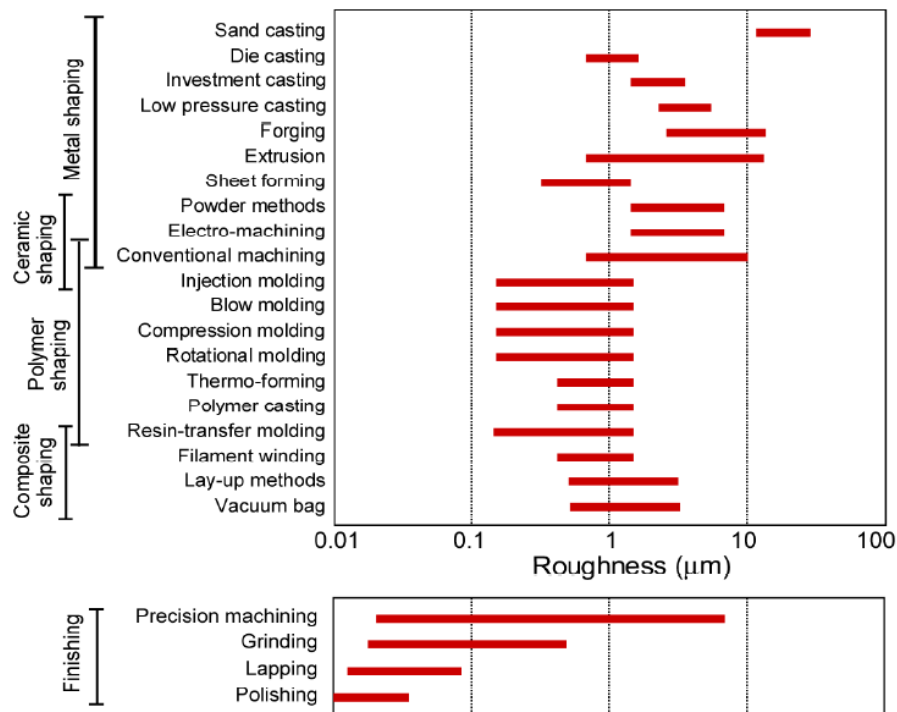
Figure D-3.   **Surface roughness range for different machining processes**
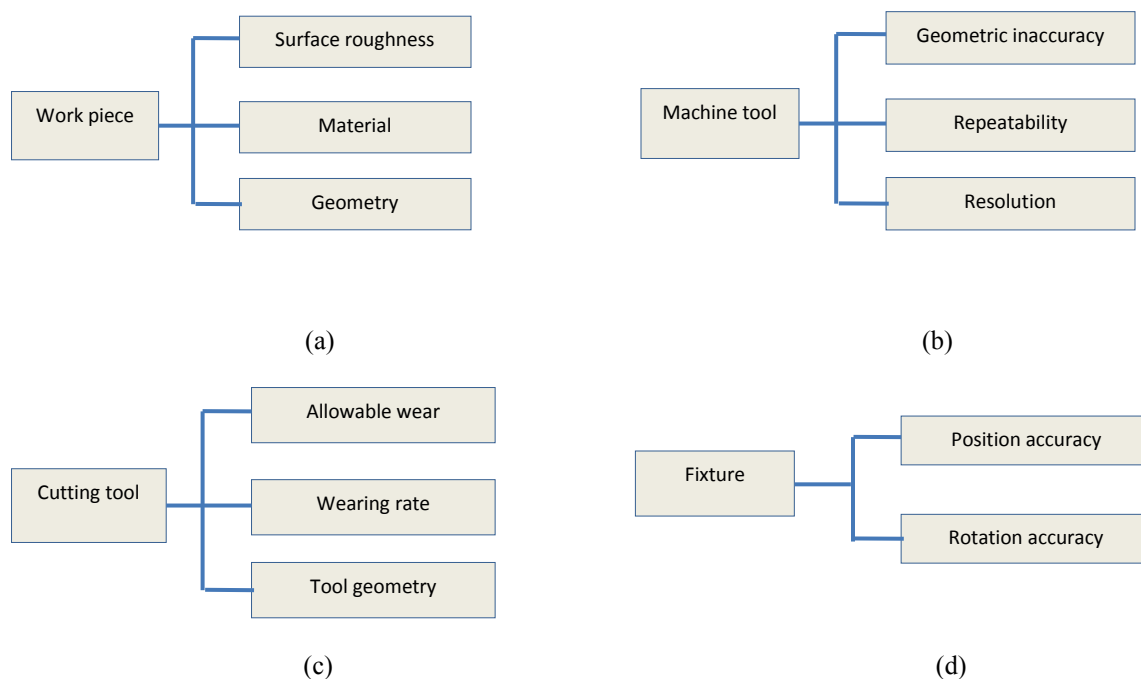

Figure D-4.   **Machining error sources. (a) Work-piece; (b) machine tool; (c) cutting tool; and (d) fixture.**

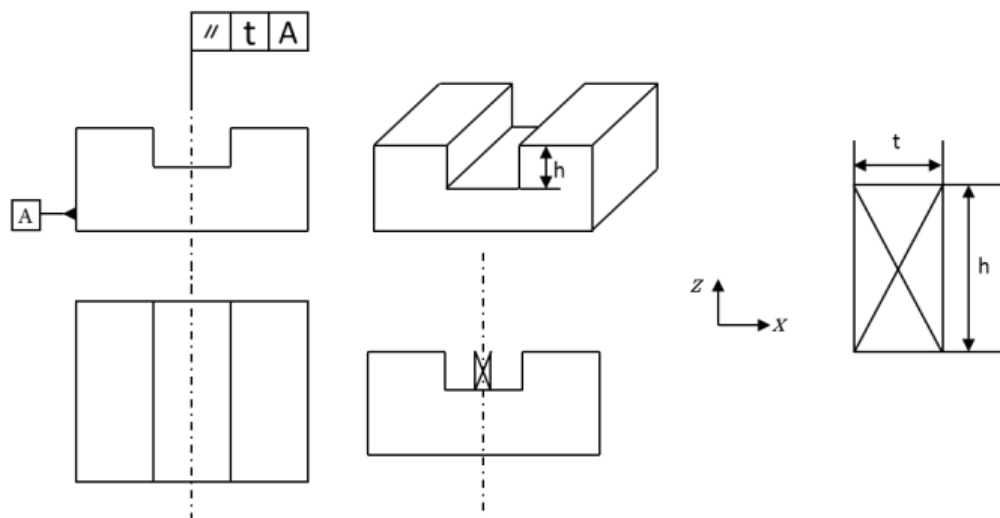**Figure D-5.**   **Parallelism tolerance for slot.**
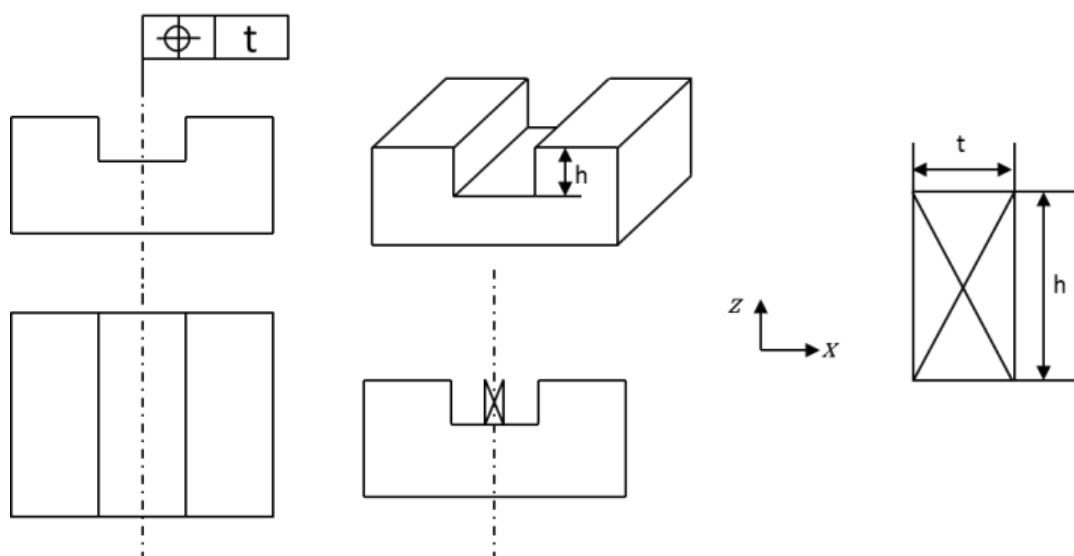
Figure D-6.   **Position tolerance for a slot.**
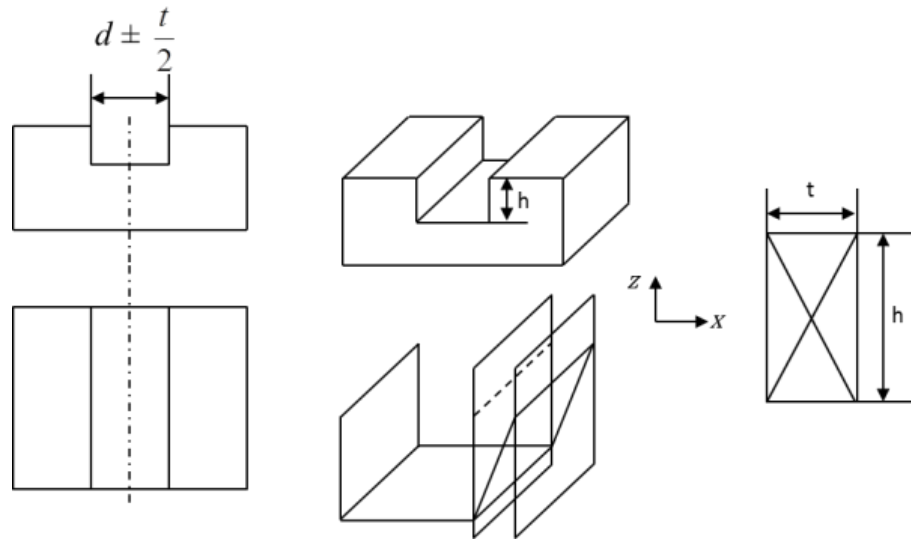
Figure D-7.   Dimension tolerance.
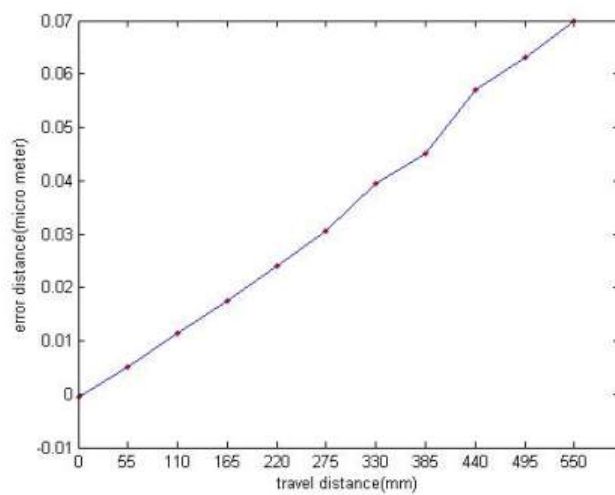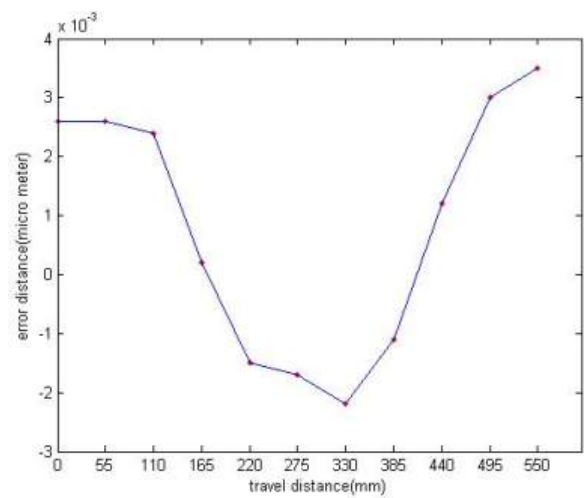
(a)   (b)
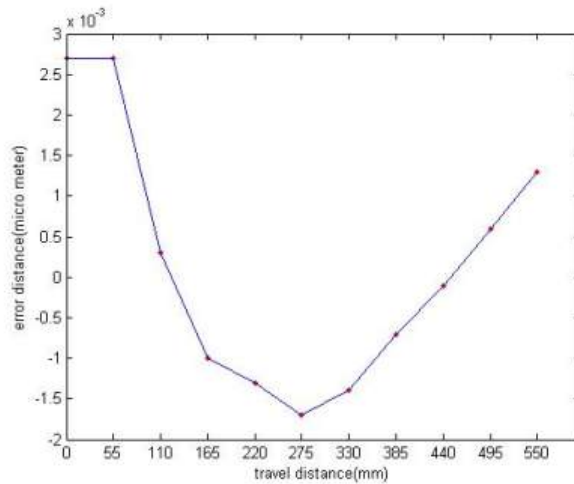
(c)

Figure D-8. Geometric errors of a single axis. (a) linear accuracy in *x* axis; (b) straightness error in *y* direction while moving in *x* direction; (c) straightness in *z* direction while moving in *x* axis.
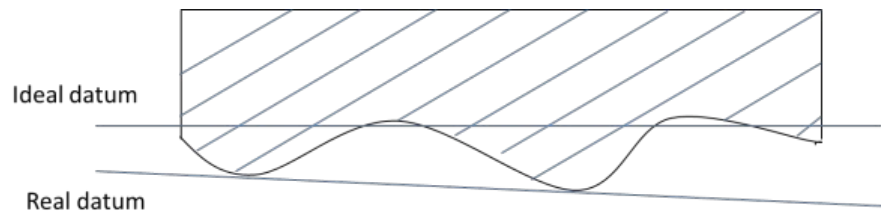
(a)
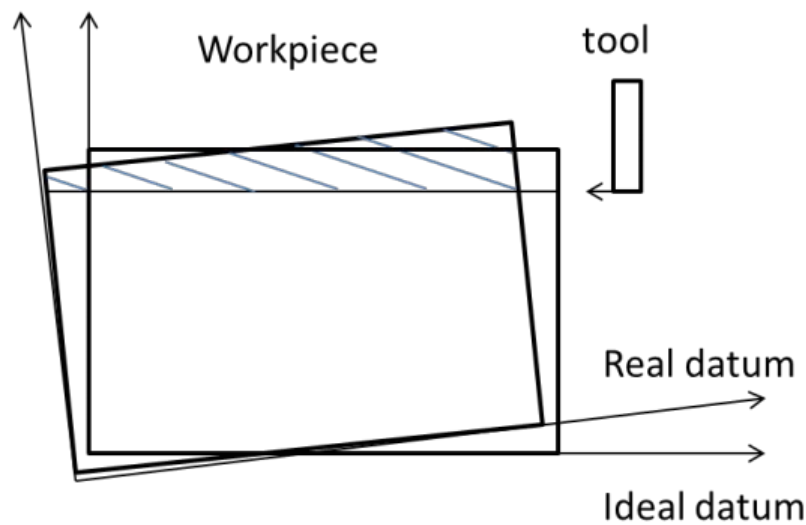
(b)

Figure D-9. Machining error from work-holding. (a) inaccuracy from work-piece surface roughness; and (b) inaccuracy from fixture setting up error
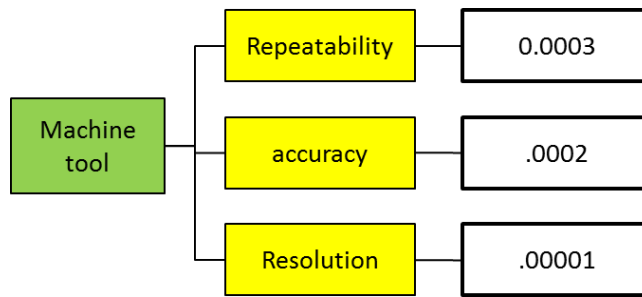
Figure D-10. **Machine tool tolerance information.**
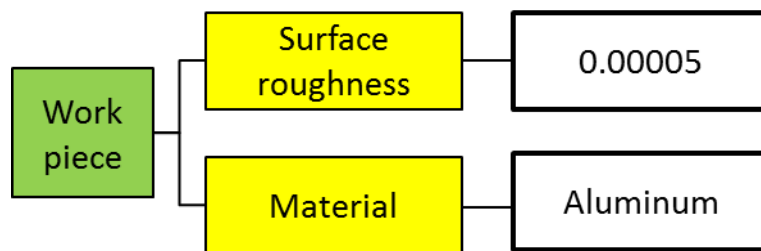
Figure D-11. **Work-piece tolerance related parameters.**
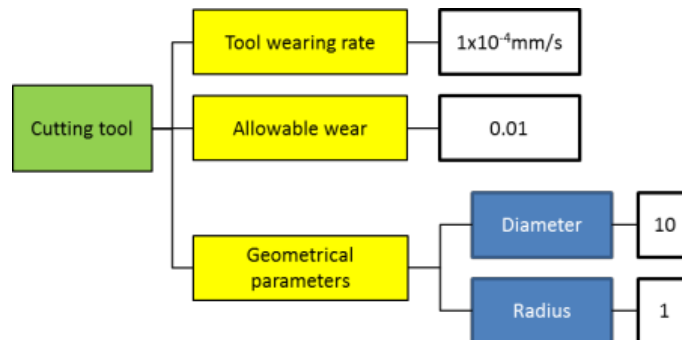
Figure D-12. **Cutting tool tolerance related parameters.**
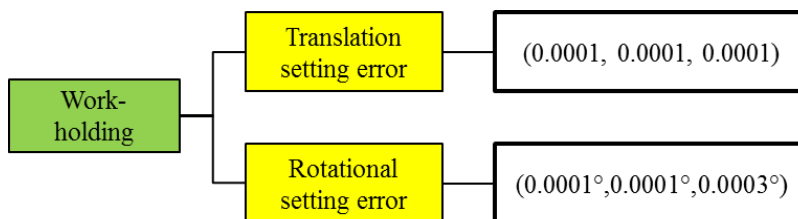
Figure D-13. **Work-holding accuracy.**

iFAB MCPML Interface Description

Version 14

9 May, 2012

Contents

1.0 INTRODUCTION

The iFAB MCPML is intended to provide information characterizing the manufacturing resources available as part of a foundry (or set of foundries) to those responding to manufacturability queries by product designers using a META toolchain and to those configuring a foundry to manufacture a design produced using a META toolchain. Accordingly, the MCPML is being developed to contain a range of information about foundry resources to support such requests. To achieve these ends, the interface to the MCPML is implemented as a set of web services that provide access to the resource descriptions and models maintained by the MCPML.

The operations provided by the MCPML web services interface fall into three main categories. Manufacturing Capability Operations that provide information about resources modeled by the MCPML. Design Specific Fabrication Operations provide information about the applicability of foundry resources toward the fabrication of a part (non-COTS design component). The Design Specific Assembly Operations provide information about the assembly of two design elements using foundry resources.

This section provides an introduction to the various operations. For each operation the name of the operation is defined, followed by its inputs and outputs, a brief description of its semantics, a pseudo API, implementation status, and sample SOAP requests and responses for most services. There is also an indication of whether the operation is a "pure data" operation, essentially just returning data from the library, or an operation with "business logic," meaning that the operation performs processing to produce the results returned. The WSDL (Web Services Description Language) file defining the programming interface including the exact interface for the individual services is provided as a separate file.

1.1 MANUFACTURING CAPABILITY OPERATIONS

These operations provide responses solely from information present in the library, and require no input that must be extracted from a design or META TDP.

1.1.1   What machines are in the library

Inputs: None

Outputs: List of machine names

Description: Returns a list of the names of the machines in the library

API: getMachineNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

Approved for public release; distribution unlimited.

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNames xmlns="http://MCPML_URI/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNamesResponse xmlns="http://MCPML_URI/">
      <getMachineNamesResult>
        <string>string</string>
        <string>string</string>
      </getMachineNamesResult>
    </getMachineNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.2   What are the main characteristics of machine M

Inputs: machine name

Outputs: Structure describing the main capabilities of the machine

Description: Return the basic machine data from the library

API: getMachineSpecifications (string machineName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineSpecifications xmlns="http://MCPML_URI/">
      <machineName>string</machineName>
    </getMachineSpecifications>
```

Approved for public release; distribution unlimited.

```
   </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineSpecificationsResponse xmlns="http://MCPML_URI/">
      <getMachineSpecificationsResult>
        <PairOfStringString>
          <attribute>string</attribute>
          <value>string</value>
        </PairOfStringString>
        <PairOfStringString>
          <attribute>string</attribute>
          <value>string</value>
        </PairOfStringString>
      </getMachineSpecificationsResult>
    </getMachineSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.3   What are the complete characteristics of machine M

Inputs: machine name

Outputs: Structure describing the main capabilities of the machine

Description: Return the basic machine data from the library

API: getFullMachineSpecifications (string machineName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFullMachineSpecifications xmlns="http://ifab.boeing.com/">
      <machineName>string</machineName>
    </getFullMachineSpecifications>
  </soap12:Body>
</soap12:Envelope>
```

Approved for public release; distribution unlimited.

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFullMachineSpecificationsResponse xmlns="http://ifab.boeing.com/">
      <getFullMachineSpecificationsResult>
        <ID>int</ID>
        <Description>string</Description>
        <Machine>string</Machine>
        <Process_1>string</Process_1>
        <Process_2>string</Process_2>
        <Process_3>string</Process_3>
        <Machine_Size_X>double</Machine_Size_X>
        <Machine_Size_Y>double</Machine_Size_Y>
        <Machine_Size_Z>double</Machine_Size_Z>
        <Table_Size_X>double</Table_Size_X>
        <Table_Size_Y>double</Table_Size_Y>
        <Table_Size_Z>double</Table_Size_Z>
        <Workpiece_Size_X>double</Workpiece_Size_X>
        <Workpiece_Size_Y>double</Workpiece_Size_Y>
        <Workpiece_Size_Z>double</Workpiece_Size_Z>
        <Workpiece_Diameter>double</Workpiece_Diameter>
        <Machine_Weight>double</Machine_Weight>
        <Max__Workpiece_Weight>double</Max__Workpiece_Weight>
        <Max_Spindle_to_Table_Distance>double</Max_Spindle_to_Table_Distance>
        <Spindle_Diameter>double</Spindle_Diameter>
        <Current_rating>double</Current_rating>
        <Voltage_rating>double</Voltage_rating>
        <Compressed_air_req>double</Compressed_air_req>
        <Noise_Level>double</Noise_Level>
        <Axes_1_Axes_Name>string</Axes_1_Axes_Name>
        <Axes_1_Dependence>string</Axes_1_Dependence>
        <Axes_1_Motion_type>string</Axes_1_Motion_type>
        <Axes_1_Motion_axis>string</Axes_1_Motion_axis>
        <Axes_1_Travel>double</Axes_1_Travel>
        <Axes_1_Travel_speed>double</Axes_1_Travel_speed>
        <Axes_2_Axes_Name>string</Axes_2_Axes_Name>
        <Axes_2_Dependence>string</Axes_2_Dependence>
        <Axes_2_Motion_type>string</Axes_2_Motion_type>
        <Axes_2_Motion_axis>string</Axes_2_Motion_axis>
        <Axes_2_Travel>double</Axes_2_Travel>
        <Axes_2_Travel_speed>double</Axes_2_Travel_speed>
        <Axes_3_Axes_Name>string</Axes_3_Axes_Name>
        <Axes_3_Dependence>string</Axes_3_Dependence>
        <Axes_3_Motion_type>string</Axes_3_Motion_type>
        <Axes_3_Motion_axis>string</Axes_3_Motion_axis>
        <Axes_3_Travel>double</Axes_3_Travel>
        <Axes_3_Angular>double</Axes_3_Angular>
        <Axes_3_Travel_speed>double</Axes_3_Travel_speed>
        <Axes_3_Rotation_Speed>double</Axes_3_Rotation_Speed>
        <Axes_4_Axes_Name>string</Axes_4_Axes_Name>
        <Axes_4_Dependence>string</Axes_4_Dependence>
```

Approved for public release; distribution unlimited.

```xml
<Axes_4_Motion_type>string</Axes_4_Motion_type>
<Axes_4_Motion_axis>string</Axes_4_Motion_axis>
<Axes_4_Travel>double</Axes_4_Travel>
<Axes_4_Angular>double</Axes_4_Angular>
<Axes_4_Travel_speed>double</Axes_4_Travel_speed>
<Axes_4_Rotation_Speed>double</Axes_4_Rotation_Speed>
<Axes_5_Axes_Name>string</Axes_5_Axes_Name>
<Axes_5_Dependence>string</Axes_5_Dependence>
<Axes_5_Motion_type>string</Axes_5_Motion_type>
<Axes_5_Motion_axis>string</Axes_5_Motion_axis>
<Axes_5_Travel>double</Axes_5_Travel>
<Axes_5_Angular>double</Axes_5_Angular>
<Axes_5_Travel_speed>double</Axes_5_Travel_speed>
<Axes_6_Axes_Name>string</Axes_6_Axes_Name>
<Axes_6_Dependence>string</Axes_6_Dependence>
<Axes_6_Motion_type>string</Axes_6_Motion_type>
<Axes_6_Motion_axis>string</Axes_6_Motion_axis>
<Axes_6_Travel>double</Axes_6_Travel>
<Axes_6_Angular>double</Axes_6_Angular>
<Axes_6_Travel_speed>double</Axes_6_Travel_speed>
<Axes_6_Rotation_Speed>double</Axes_6_Rotation_Speed>
<Axes_7_Axes_Name>string</Axes_7_Axes_Name>
<Axes_7_Dependence>string</Axes_7_Dependence>
<Axes_7_Motion_type>string</Axes_7_Motion_type>
<Axes_7_Motion_axis>string</Axes_7_Motion_axis>
<Axes_7_Travel>double</Axes_7_Travel>
<Axes_7_Travel_speed>double</Axes_7_Travel_speed>
<Axes_8_Axes_Name>string</Axes_8_Axes_Name>
<Axes_8_Dependence>string</Axes_8_Dependence>
<Axes_8_Motion_type>string</Axes_8_Motion_type>
<Axes_8_Motion_axis>string</Axes_8_Motion_axis>
<Axes_8_Travel>double</Axes_8_Travel>
<Axes_8_Travel_speed>double</Axes_8_Travel_speed>
<Axes_9_Axes_Name>string</Axes_9_Axes_Name>
<Axes_9_Dependence>string</Axes_9_Dependence>
<Axes_9_Motion_type>string</Axes_9_Motion_type>
<Axes_9_Motion_axis>string</Axes_9_Motion_axis>
<Axes_9_Travel>double</Axes_9_Travel>
<Axes_9_Travel_speed>double</Axes_9_Travel_speed>
<Axes_10_Axes_Name>string</Axes_10_Axes_Name>
<Axes_10_Dependence>string</Axes_10_Dependence>
<Axes_10_Motion_type>string</Axes_10_Motion_type>
<Axes_10_Motion_axis>string</Axes_10_Motion_axis>
<Axes_10_Travel>double</Axes_10_Travel>
<Axes_10_Travel_speed>double</Axes_10_Travel_speed>
<Motion_Repeatability>double</Motion_Repeatability>
<Motion_Accuracy>double</Motion_Accuracy>
<Motion_Resolution>double</Motion_Resolution>
<Load_X___Axis>double</Load_X___Axis>
<Load_Y___Axis>double</Load_Y___Axis>
<Load_Z___Axis>double</Load_Z___Axis>
<_Spindle_speed>int</_Spindle_speed>
<Spindle_speed_Min>int</Spindle_speed_Min>
<Spindle_speed_Max>int</Spindle_speed_Max>
<Power>double</Power>
<Number_of_Tools>int</Number_of_Tools>
<T1_>string</T1_>
```

```
        <T2>string</T2>
        <T3>string</T3>
        <T4>string</T4>
        <Module_Name>string</Module_Name>
        <Tool_Change_Time>double</Tool_Change_Time>
        <Maintenance_Info>string</Maintenance_Info>
        <Name__Serial_Number_>string</Name__Serial_Number_>
        <Vendor_Info>string</Vendor_Info>
        <Operator_Info>string</Operator_Info>
        <Manufacture_Date>string</Manufacture_Date>
        <Lubricant_>string</Lubricant_>
        <Coolant>string</Coolant>
        <Billing_rate>double</Billing_rate>
        <Purchase_price>double</Purchase_price>
        <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
        <length_of_the_loan>double</length_of_the_loan>
        <Annual_maintenance_costs>double</Annual_maintenance_costs>
        <Price_of_electricity>double</Price_of_electricity>
        <Projected_machine_use>double</Projected_machine_use>
        <cost_of_consumables>double</cost_of_consumables>
        <Depreciation_Period>double</Depreciation_Period>
        <Max_tool_length>string</Max_tool_length>
        <Control>string</Control>
        <Company>string</Company>
      </getFullMachineSpecificationsResult>
    </getFullMachineSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.4   What processes are in the library

Inputs: None

Outputs: List of processes

Description: Returns the list of processes in the library

API: getProcessNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getProcessNames xmlns="http://MCPML_URI/" />
  </soap12:Body>
```

Approved for public release; distribution unlimited.

```
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getProcessNamesResponse xmlns="http://MCPML_URI/">
      <getProcessNamesResult>
        <string>string</string>
        <string>string</string>
      </getProcessNamesResult>
    </getProcessNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.5   What are the specifications of Process P

Inputs: process name

Outputs: process information

Description: return basic process data from the library

API: getProcessSpecifications(string processName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getProcessSpecifications xmlns="http://MCPML_URI/">
      <processName>string</processName>
    </getProcessSpecifications>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

Approved for public release; distribution unlimited.

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getProcessSpecificationsResponse xmlns="http://MCPML_URI/">
      <getProcessSpecificationsResult>
        <ProcessSpec>
          <name>string</name>
          <description>string</description>
          <activityId>int</activityId>
          <processSteps>
            <procStruct xsi:nil="true" />
            <procStruct xsi:nil="true" />
          </processSteps>
        </ProcessSpec>
        <ProcessSpec>
          <name>string</name>
          <description>string</description>
          <activityId>int</activityId>
          <processSteps>
            <procStruct xsi:nil="true" />
            <procStruct xsi:nil="true" />
          </processSteps>
        </ProcessSpec>
      </getProcessSpecificationsResult>
    </getProcessSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.6   What fastening methods are supported by the library

Inputs: None

Outputs: List of fastening methods

Description: Return the fastening methods supported by the library. Currently defined by the fastening methods of the CyPhy ML metamodel.

API: getFasteningMethods ()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFasteningMethods xmlns="http://MCPML_URI/" />
```

Approved for public release; distribution unlimited.

```
   </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFasteningMethodsResponse xmlns="http://MCPML_URI/">
      <getFasteningMethodsResult>
        <string>string</string>
        <string>string</string>
      </getFasteningMethodsResult>
    </getFasteningMethodsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.7   What fastening mediums are supported by the library

Inputs: Fastening Method

Outputs: List of fastening mediums

Description: Returns the fastening mediums relevant to the fastening method. Currently defined by the fastener types in the CyPhy ML metamodel. See 1.1.6 for the list of available mechanisms.

API: getFasteningMedium (string FasteningMethod)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFasteningMedium xmlns="http://MCPML_URI/">
      <FasteningMethod>string</FasteningMethod>
    </getFasteningMedium>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

Approved for public release; distribution unlimited.

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getFasteningMediumResponse xmlns="http://MCPML_URI/">
      <getFasteningMediumResult>
        <string>string</string>
        <string>string</string>
      </getFasteningMediumResult>
    </getFasteningMediumResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.8   What tooling is in the library

Inputs: None

Outputs: List of tooling

Description: Returns the list of tooling in the library

API: getToolingNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNamesResponse xmlns="http://ifab.boeing.com/">
      <getToolingNamesResult>
        <string>string</string>
        <string>string</string>
```

Approved for public release; distribution unlimited.

```
        </getToolingNamesResult>
      </getToolingNamesResponse>
    </soap12:Body>
</soap12:Envelope>
```

1.1.9 What are the specifications of tooling T

Inputs: tooling name

Outputs: tooling information

Description: return basic tooling data from the library

API: getToolingSpecification (string toolingName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingSpecification xmlns="http://ifab.boeing.com/">
      <ToolingName>string</ToolingName>
    </getToolingSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingSpecificationResponse xmlns="http://ifab.boeing.com/">
      <getToolingSpecificationResult>
        <toolingSpec>
          <tooling>
            <ID>int</ID>
            <Tool>string</Tool>
            <Overall_Length>double</Overall_Length>
            <Shank_diameter>double</Shank_diameter>
            <No__of_flutes>double</No__of_flutes>
            <Helix_angle>double</Helix_angle>
            <Corner_radius>string</Corner_radius>
            <Reach_length>double</Reach_length>
```

```xml
    <Tool_material>string</Tool_material>
    <Coat_material>string</Coat_material>
    <Tool_hardness>double</Tool_hardness>
    <Optimum_tool_life>string</Optimum_tool_life>
    <TOLERANCE_MODEL>double</TOLERANCE_MODEL>
    <Ordering_number>double</Ordering_number>
    <Vendor_info>string</Vendor_info>
    <Serial_number>string</Serial_number>
    <Manufacturing_date>string</Manufacturing_date>
    <Price_of_tool>double</Price_of_tool>
    <Tool_usage>double</Tool_usage>
    <Maintenance_costs>double</Maintenance_costs>
</tooling>
<proc>
    <ID>int</ID>
    <Name>string</Name>
</proc>
<toolingMaterial>
    <Tooling>int</Tooling>
    <Material>int</Material>
    <Optimal_feed_min>double</Optimal_feed_min>
    <Optimal_feed_max>double</Optimal_feed_max>
    <Optimal_speed>string</Optimal_speed>
    <Chipload_per_flute>double</Chipload_per_flute>
    <Coolant>int</Coolant>
</toolingMaterial>
<coolant>
    <ID>int</ID>
    <Name>string</Name>
</coolant>
<material>
    <ID>int</ID>
    <Material1>string</Material1>
    <Alloy_Class>string</Alloy_Class>
    <Alloy>double</Alloy>
    <Identifier>string</Identifier>
    <Common_Name>string</Common_Name>
    <Description>string</Description>
    <Tensile_Strength>double</Tensile_Strength>
    <Yield_Strength>double</Yield_Strength>
    <Brinell_Hardness>double</Brinell_Hardness>
    <Rockwell_Hardness>string</Rockwell_Hardness>
    <Elongation_Minimum>string</Elongation_Minimum>
    <Elongation_maximum>double</Elongation_maximum>
    <Machinability>double</Machinability>
    <Electrical_Conductivity>double</Electrical_Conductivity>
    <Aluminum__Al__Minimum>double</Aluminum__Al__Minimum>
    <Aluminum__Al___Maximum>double</Aluminum__Al___Maximum>
    <Bismuth__Bi__Minimum>double</Bismuth__Bi__Minimum>
    <Bismuth__Bi__Maximum>double</Bismuth__Bi__Maximum>
    <Chromium__Cr___Minimum>double</Chromium__Cr___Minimum>
    <Chromium__Cr___Maximum>double</Chromium__Cr___Maximum>
    <Copper__Cu___Minimum>double</Copper__Cu___Minimum>
    <Copper__Cu___Maximum>double</Copper__Cu___Maximum>
    <Iron__Fe___Minimum>string</Iron__Fe___Minimum>
    <Iron__Fe___Maximum>double</Iron__Fe___Maximum>
    <Lead__Pb__Minimum>double</Lead__Pb__Minimum>
```

```xml
      <Lead__Pb__Maximum>double</Lead__Pb__Maximum>
      <Magnesium__Mg___Minimum>double</Magnesium__Mg___Minimum>
      <Magnesium__Mg__Maximum>double</Magnesium__Mg__Maximum>
      <Manganese__Mn__Minimum>double</Manganese__Mn__Minimum>
      <Manganese__Mn__Maximum>double</Manganese__Mn__Maximum>
      <Silicon__Si___Minimum>double</Silicon__Si___Minimum>
      <Silicon__Si___Maximum>double</Silicon__Si___Maximum>
      <Zinc__Zn__Minimum>double</Zinc__Zn__Minimum>
      <Zinc__Zn__Maximum>double</Zinc__Zn__Maximum>
      <Boron_Minimum>string</Boron_Minimum>
      <Boron_Maximum>double</Boron_Maximum>
      <Carbon_Minimum>string</Carbon_Minimum>
      <Carbon_Maximum>string</Carbon_Maximum>
      <Phosphorous_Minimum>string</Phosphorous_Minimum>
      <Phosphorous_Maximum>string</Phosphorous_Maximum>
      <Sulfur_Minimum>string</Sulfur_Minimum>
      <Sulfur_Maximum>string</Sulfur_Maximum>
    </material>
  </toolingSpec>
  <toolingSpec>
    <tooling>
      <ID>int</ID>
      <Tool>string</Tool>
      <Overall_Length>double</Overall_Length>
      <Shank_diameter>double</Shank_diameter>
      <No__of_flutes>double</No__of_flutes>
      <Helix_angle>double</Helix_angle>
      <Corner_radius>string</Corner_radius>
      <Reach_length>double</Reach_length>
      <Tool_material>string</Tool_material>
      <Coat_material>string</Coat_material>
      <Tool_hardness>double</Tool_hardness>
      <Optimum_tool_life>string</Optimum_tool_life>
      <TOLERANCE_MODEL>double</TOLERANCE_MODEL>
      <Ordering_number>double</Ordering_number>
      <Vendor_info>string</Vendor_info>
      <Serial_number>string</Serial_number>
      <Manufacturing_date>string</Manufacturing_date>
      <Price_of_tool>double</Price_of_tool>
      <Tool_usage>double</Tool_usage>
      <Maintenance_costs>double</Maintenance_costs>
    </tooling>
    <proc>
      <ID>int</ID>
      <Name>string</Name>
    </proc>
    <toolingMaterial>
      <Tooling>int</Tooling>
      <Material>int</Material>
      <Optimal_feed_min>double</Optimal_feed_min>
      <Optimal_feed_max>double</Optimal_feed_max>
      <Optimal_speed>string</Optimal_speed>
      <Chipload_per_flute>double</Chipload_per_flute>
      <Coolant>int</Coolant>
    </toolingMaterial>
    <coolant>
      <ID>int</ID>
```

```xml
            <Name>string</Name>
          </coolant>
          <material>
            <ID>int</ID>
            <Material1>string</Material1>
            <Alloy_Class>string</Alloy_Class>
            <Alloy>double</Alloy>
            <Identifier>string</Identifier>
            <Common_Name>string</Common_Name>
            <Description>string</Description>
            <Tensile_Strength>double</Tensile_Strength>
            <Yield_Strength>double</Yield_Strength>
            <Brinell_Hardness>double</Brinell_Hardness>
            <Rockwell_Hardness>string</Rockwell_Hardness>
            <Elongation_Minimum>string</Elongation_Minimum>
            <Elongation_maximum>double</Elongation_maximum>
            <Machinability>double</Machinability>
            <Electrical_Conductivity>double</Electrical_Conductivity>
            <Aluminum__Al__Minimum>double</Aluminum__Al__Minimum>
            <Aluminum__Al___Maximum>double</Aluminum__Al___Maximum>
            <Bismuth__Bi__Minimum>double</Bismuth__Bi__Minimum>
            <Bismuth__Bi__Maximum>double</Bismuth__Bi__Maximum>
            <Chromium__Cr___Minimum>double</Chromium__Cr___Minimum>
            <Chromium__Cr___Maximum>double</Chromium__Cr___Maximum>
            <Copper__Cu___Minimum>double</Copper__Cu___Minimum>
            <Copper__Cu___Maximum>double</Copper__Cu___Maximum>
            <Iron__Fe___Minimum>string</Iron__Fe___Minimum>
            <Iron__Fe___Maximum>double</Iron__Fe___Maximum>
            <Lead__Pb__Minimum>double</Lead__Pb__Minimum>
            <Lead__Pb__Maximum>double</Lead__Pb__Maximum>
            <Magnesium__Mg___Minimum>double</Magnesium__Mg___Minimum>
            <Magnesium__Mg__Maximum>double</Magnesium__Mg__Maximum>
            <Manganese__Mn__Minimum>double</Manganese__Mn__Minimum>
            <Manganese__Mn__Maximum>double</Manganese__Mn__Maximum>
            <Silicon__Si___Minimum>double</Silicon__Si___Minimum>
            <Silicon__Si___Maximum>double</Silicon__Si___Maximum>
            <Zinc__Zn__Minimum>double</Zinc__Zn__Minimum>
            <Zinc__Zn__Maximum>double</Zinc__Zn__Maximum>
            <Boron_Minimum>string</Boron_Minimum>
            <Boron_Maximum>double</Boron_Maximum>
            <Carbon_Minimum>string</Carbon_Minimum>
            <Carbon_Maximum>string</Carbon_Maximum>
            <Phosphorous_Minimum>string</Phosphorous_Minimum>
            <Phosphorous_Maximum>string</Phosphorous_Maximum>
            <Sulfur_Minimum>string</Sulfur_Minimum>
            <Sulfur_Maximum>string</Sulfur_Maximum>
          </material>
        </toolingSpec>
      </getToolingSpecificationResult>
    </getToolingSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

Approved for public release; distribution unlimited.

1.1.10   What tools are in the library

Inputs: None

Outputs: List of tools

Description: Returns the list of (hand) tools in the library

API: getHandToolNames ()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNamesResponse xmlns="http://ifab.boeing.com/">
      <getHandToolNamesResult>
        <string>string</string>
        <string>string</string>
      </getHandToolNamesResult>
    </getHandToolNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.11   What are the specifications of tool T

Inputs: tool name

Outputs: tool information

Description: return basic (hand) tool data from the library

API: getHandToolSpecifications (string handToolName)

Approved for public release; distribution unlimited.

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolSpecifications xmlns="http://ifab.boeing.com/">
      <ToolName>string</ToolName>
    </getHandToolSpecifications>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolSpecificationsResponse xmlns="http://ifab.boeing.com/">
      <getHandToolSpecificationsResult>
        <Id>int</Id>
        <Name>string</Name>
        <Hand_tool>string</Hand_tool>
        <Process_1>string</Process_1>
        <Mounting___Spline>string</Mounting___Spline>
        <Weight_excluding_chuck_collet>double</Weight_excluding_chuck_collet>
        <Weight_including_chuck_collet>double</Weight_including_chuck_collet>
        <Chuck_collet_size>string</Chuck_collet_size>
        <Air_hose_inlet_size>double</Air_hose_inlet_size>
        <Min_socket>string</Min_socket>
        <Max_socket>string</Max_socket>
        <Cutting_pad_diameter>double</Cutting_pad_diameter>
        <Alumina_nozzle>double</Alumina_nozzle>
        <AC_Voltage>double</AC_Voltage>
        <AC_Current>double</AC_Current>
        <Battery_Type>string</Battery_Type>
        <Battery_Voltage>string</Battery_Voltage>
        <Battery_Capacity>string</Battery_Capacity>

<Air_Consumption_rate_at_max_output>double</Air_Consumption_rate_at_max_outpu
t>

<Air_Consumption_rate_at_free_speed>double</Air_Consumption_rate_at_free_spee
d>

<Pnuematic_recommended_hose_size>double</Pnuematic_recommended_hose_size>
```

```
        <Operating_temperature_min>double</Operating_temperature_min>
        <Operating_temperature_max>double</Operating_temperature_max>
        <Sound_level>double</Sound_level>
        <Max_power_output>double</Max_power_output>
        <Max_operating_pressure>double</Max_operating_pressure>
        <Measured_vibration_value>double</Measured_vibration_value>
        <Max_cutting_depth>string</Max_cutting_depth>
        <Setting_range>double</Setting_range>
        <Duty_Cycle>double</Duty_Cycle>
        <Rated_Amps>double</Rated_Amps>
        <Input_Power>string</Input_Power>
        <Input_Current_at_Rated_Output>string</Input_Current_at_Rated_Output>
        <Rated_Output>string</Rated_Output>
        <Torque_Min>double</Torque_Min>
        <Torque_Max>double</Torque_Max>
        <Max_Torque>string</Max_Torque>
        <Speed_at_max_output>string</Speed_at_max_output>
        <Speed_at_free_speed>double</Speed_at_free_speed>
        <Impacts_min>string</Impacts_min>
        <Gear_ratio>string</Gear_ratio>
        <Length_feed>string</Length_feed>
        <Stroke>string</Stroke>
        <CS_distance>double</CS_distance>
        <Tool_Envelope_Length>double</Tool_Envelope_Length>
        <Tool_Envelope_Width>double</Tool_Envelope_Width>
        <Tool_Envelope_Height>double</Tool_Envelope_Height>
        <Height_over_spindle_D>string</Height_over_spindle_D>
        <Height_over_spindle_T>string</Height_over_spindle_T>
        <Height_over_spindle_H>double</Height_over_spindle_H>
        <Angle_head_height_>double</Angle_head_height_>
        <Chuck_Collet_T1>string</Chuck_Collet_T1>
        <Chuck_Collet_T2>string</Chuck_Collet_T2>
        <Hook_up_Kit_Length>double</Hook_up_Kit_Length>
        <Hook_Up_Kit_Width_>double</Hook_Up_Kit_Width_>
        <Hook_up_Kit_Height>double</Hook_up_Kit_Height>
        <Hook_up_Kit_Weight>double</Hook_up_Kit_Weight>
        <Drive_Square_>string</Drive_Square_>
        <Drive_Hexa>string</Drive_Hexa>
        <Drive_Ratchet>string</Drive_Ratchet>
        <Model_number>string</Model_number>
        <Hand_tool1>string</Hand_tool1>
        <Abrasive_wheel>string</Abrasive_wheel>
        <Service_kit>string</Service_kit>
        <Vendor_info>string</Vendor_info>
        <Manufacturing_date>dateTime</Manufacturing_date>
        <CAD_Drawing_file>string</CAD_Drawing_file>
        <Billing_rate>double</Billing_rate>
        <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
        <Annual_maintenance_costs>double</Annual_maintenance_costs>
        <Price_of_electricity___KW>double</Price_of_electricity___KW>

<Projected_machine_hours___year>double</Projected_machine_hours___year>
        <cost_of_consumables>double</cost_of_consumables>
        <Price>double</Price>
      </getHandToolSpecificationsResult>
    </getHandToolSpecificationsResponse>
  </soap12:Body>
```

```
</soap12:Envelope>
```

1.1.12   What are the tool types for activity A

Inputs: Activity

Outputs: List of tool types

Description: Returns the list of tool types usable in performing activity A. The tool type is defined in the HAML.

API: getToolTypesForActivity (string toolType)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolTypesForActivity xmlns="http://ifab.boeing.com/">
      <activity>string</activity>
    </getToolTypesForActivity>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolTypesForActivityResponse xmlns="http://ifab.boeing.com/">
      <getToolTypesForActivityResult>
        <string>string</string>
        <string>string</string>
      </getToolTypesForActivityResult>
    </getToolTypesForActivityResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.13   What materials are in the library

Inputs: None

Outputs: List of materials

Description: Returns the list of materials in the library

API: getMaterialNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialNamesResponse xmlns="http://ifab.boeing.com/">
      <getMaterialNamesResult>
        <string>string</string>
        <string>string</string>
      </getMaterialNamesResult>
    </getMaterialNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.14   What are the specifications of material M

Inputs: material name

Outputs: material information

Description: return basic material information from the library

API: getMaterialSpecifications(string materialName)

Approved for public release; distribution unlimited.

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialSpecifications xmlns="http://ifab.boeing.com/">
      <materialIdentifier>string</materialIdentifier>
    </getMaterialSpecifications>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMaterialSpecificationsResponse xmlns="http://ifab.boeing.com/">
      <getMaterialSpecificationsResult>
        <ID>int</ID>
        <Material1>string</Material1>
        <Alloy_Class>string</Alloy_Class>
        <Alloy>double</Alloy>
        <Identifier>string</Identifier>
        <Common_Name>string</Common_Name>
        <Description>string</Description>
        <Tensile_Strength>double</Tensile_Strength>
        <Yield_Strength>double</Yield_Strength>
        <Brinell_Hardness>double</Brinell_Hardness>
        <Rockwell_Hardness>string</Rockwell_Hardness>
        <Elongation_Minimum>string</Elongation_Minimum>
        <Elongation_maximum>double</Elongation_maximum>
        <Machinability>double</Machinability>
        <Electrical_Conductivity>double</Electrical_Conductivity>
        <Aluminum__Al__Minimum>double</Aluminum__Al__Minimum>
        <Aluminum__Al___Maximum>double</Aluminum__Al___Maximum>
        <Bismuth__Bi__Minimum>double</Bismuth__Bi__Minimum>
        <Bismuth__Bi__Maximum>double</Bismuth__Bi__Maximum>
        <Chromium__Cr___Minimum>double</Chromium__Cr___Minimum>
        <Chromium__Cr___Maximum>double</Chromium__Cr___Maximum>
        <Copper__Cu___Minimum>double</Copper__Cu___Minimum>
        <Copper__Cu___Maximum>double</Copper__Cu___Maximum>
        <Iron__Fe___Minimum>string</Iron__Fe___Minimum>
        <Iron__Fe___Maximum>double</Iron__Fe___Maximum>
        <Lead__Pb__Minimum>double</Lead__Pb__Minimum>
```

```
        <Lead__Pb__Maximum>double</Lead__Pb__Maximum>
        <Magnesium__Mg___Minimum>double</Magnesium__Mg___Minimum>
        <Magnesium__Mg__Maximum>double</Magnesium__Mg__Maximum>
        <Manganese__Mn__Minimum>double</Manganese__Mn__Minimum>
        <Manganese__Mn__Maximum>double</Manganese__Mn__Maximum>
        <Silicon__Si___Minimum>double</Silicon__Si___Minimum>
        <Silicon__Si___Maximum>double</Silicon__Si___Maximum>
        <Zinc__Zn__Minimum>double</Zinc__Zn__Minimum>
        <Zinc__Zn__Maximum>double</Zinc__Zn__Maximum>
        <Boron_Minimum>string</Boron_Minimum>
        <Boron_Maximum>double</Boron_Maximum>
        <Carbon_Minimum>string</Carbon_Minimum>
        <Carbon_Maximum>string</Carbon_Maximum>
        <Phosphorous_Minimum>string</Phosphorous_Minimum>
        <Phosphorous_Maximum>string</Phosphorous_Maximum>
        <Sulfur_Minimum>string</Sulfur_Minimum>
        <Sulfur_Maximum>string</Sulfur_Maximum>
        <GUID>guid</GUID>
        <Source>string</Source>
      </getMaterialSpecificationsResult>
    </getMaterialSpecificationsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.15   What resources are in the library

Inputs: None

Outputs: List of resources

Description: Returns the list of resources in the library

API: getResourceNames()

Status: In Development

Type: Pure data

1.1.16   What are the specifications of Resource R

Inputs: resource name

Outputs: resource information

Description: return basic resource information from the library

API: getResourceSpecifications(string resourceName)

Status: In Development

Type: Pure data

1.1.17   What operator certifications are in the library

Inputs: None

Outputs: List of certifications

Description: Returns the list of operator certifications in the library

API: getCertificateNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateNamesResponse xmlns="http://ifab.boeing.com/">
      <getCertificateNamesResult>
        <string>string</string>
        <string>string</string>
      </getCertificateNamesResult>
    </getCertificateNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.18   What is the specification of certification C

Inputs: certification name

Outputs: certification information

Description: return basic certification information from the library. Each certification defines a set of processes the holder of the certification is qualified to perform, with an optional

Approved for public release; distribution unlimited.

proficiency rating, and a set of machines the holder of the certification is qualified to operate, also with an optional proficiency.

API: getCertificateSpecification (string certificationName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateSpecification xmlns="http://ifab.boeing.com/">
      <Name>string</Name>
    </getCertificateSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCertificateSpecificationResponse xmlns="http://ifab.boeing.com/">
      <getCertificateSpecificationResult>
        <Certification>string</Certification>
        <Description>string</Description>
        <ProcessProficiency>
          <Certification_Process>
            <Certification>string</Certification>
            <Process>string</Process>
            <Proficiency>double</Proficiency>
          </Certification_Process>
          <Certification_Process>
            <Certification>string</Certification>
            <Process>string</Process>
            <Proficiency>double</Proficiency>
          </Certification_Process>
        </ProcessProficiency>
        <MachineProficiency>
          <Certification_Machine>
            <Certification>string</Certification>
            <Machine>string</Machine>
            <Proficiency>double</Proficiency>
          </Certification_Machine>
```

Approved for public release; distribution unlimited.

```
        <Certification_Machine>
          <Certification>string</Certification>
          <Machine>string</Machine>
          <Proficiency>double</Proficiency>
        </Certification_Machine>
      </MachineProficiency>
    </getCertificateSpecificationResult>
  </getCertificateSpecificationResponse>
 </soap12:Body>
</soap12:Envelope>
```

1.1.19   What Bill of Process process types are in the library

Inputs: None

Outputs: List of bill of process process types

Description: Returns the list of process process types in the library

API: getBillOfProcessTypes()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBillOfProcessTypes xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBillOfProcessTypesResponse xmlns="http://ifab.boeing.com/">
      <getBillOfProcessTypesResult>
        <string>string</string>
        <string>string</string>
      </getBillOfProcessTypesResult>
    </getBillOfProcessTypesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Approved for public release; distribution unlimited.

1.1.20　What Bill of Process processes are in the library

Inputs: None

Outputs: List of bill of process process names.

Description: Returns the list of bill of process processes in the library

API: getBillOfProcessNames()

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBillOfProcessNames xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBillOfProcessNamesResponse xmlns="http://ifab.boeing.com/">
      <getBillOfProcessNamesResult>
        <string>string</string>
        <string>string</string>
      </getBillOfProcessNamesResult>
    </getBillOfProcessNamesResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.21　What is the URI for element with GUID G

Inputs: GUID

Outputs: URI for the library element.

Description: Returns the URI for the library element with the given GUID.

API: getURIforGuid (guid G)

Approved for public release; distribution unlimited.

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getURIforGuid xmlns="http://ifab.boeing.com/">
      <guid>guid</guid>
    </getURIforGuid>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getURIforGuidResponse xmlns="http://ifab.boeing.com/">
      <getURIforGuidResult>string</getURIforGuidResult>
    </getURIforGuidResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.1.22   What is the URI for the element with GUID string S

Inputs: GUID string

Outputs: URI for the library element.

Description: Returns the URI for the library element with the given GUID in string form.

API: getURIforGuid (string G)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

Approved for public release; distribution unlimited.

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getURIforGUIDString xmlns="http://ifab.boeing.com/">
      <guidString>string</guidString>
    </getURIforGUIDString>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getURIforGUIDStringResponse xmlns="http://ifab.boeing.com/">
      <getURIforGUIDStringResult>string</getURIforGUIDStringResult>
    </getURIforGUIDStringResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.2 LIBRARY UPDATE OPERATIONS

The library interface provides functions for adding and deleting library elements. The delete operations use IDs to avoid ambiguity, so there are also services for listing all IDs for the library elements. These services (especially the delete operations) should be used with caution. We may provide additional access control mechanisms for these services.

1.2.1   Add machine M to the library

Inputs: Machine specification

Output: Boolean

Description: Add the machine.

API: setMachineSpecification (<machine specification>)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
```

Approved for public release; distribution unlimited.

```xml
<setMachineSpecification xmlns="http://ifab.boeing.com/">
  <md>
    <ID>int</ID>
    <Description>string</Description>
    <Machine>string</Machine>
    <Process_1>string</Process_1>
    <Process_2>string</Process_2>
    <Process_3>string</Process_3>
    <Machine_Size_X>double</Machine_Size_X>
    <Machine_Size_Y>double</Machine_Size_Y>
    <Machine_Size_Z>double</Machine_Size_Z>
    <Table_Size_X>double</Table_Size_X>
    <Table_Size_Y>double</Table_Size_Y>
    <Table_Size_Z>double</Table_Size_Z>
    <Workpiece_Size_X>double</Workpiece_Size_X>
    <Workpiece_Size_Y>double</Workpiece_Size_Y>
    <Workpiece_Size_Z>double</Workpiece_Size_Z>
    <Workpiece_Diameter>double</Workpiece_Diameter>
    <Machine_Weight>double</Machine_Weight>
    <Max__Workpiece_Weight>double</Max__Workpiece_Weight>
    <Max_Spindle_to_Table_Distance>double</Max_Spindle_to_Table_Distance>
    <Spindle_Diameter>double</Spindle_Diameter>
    <Current_rating>double</Current_rating>
    <Voltage_rating>double</Voltage_rating>
    <Compressed_air_req>double</Compressed_air_req>
    <Noise_Level>double</Noise_Level>
    <Axes_1_Axes_Name>string</Axes_1_Axes_Name>
    <Axes_1_Dependence>string</Axes_1_Dependence>
    <Axes_1_Motion_type>string</Axes_1_Motion_type>
    <Axes_1_Motion_axis>string</Axes_1_Motion_axis>
    <Axes_1_Travel>double</Axes_1_Travel>
    <Axes_1_Travel_speed>double</Axes_1_Travel_speed>
    <Axes_2_Axes_Name>string</Axes_2_Axes_Name>
    <Axes_2_Dependence>string</Axes_2_Dependence>
    <Axes_2_Motion_type>string</Axes_2_Motion_type>
    <Axes_2_Motion_axis>string</Axes_2_Motion_axis>
    <Axes_2_Travel>double</Axes_2_Travel>
    <Axes_2_Travel_speed>double</Axes_2_Travel_speed>
    <Axes_3_Axes_Name>string</Axes_3_Axes_Name>
    <Axes_3_Dependence>string</Axes_3_Dependence>
    <Axes_3_Motion_type>string</Axes_3_Motion_type>
    <Axes_3_Motion_axis>string</Axes_3_Motion_axis>
    <Axes_3_Travel>double</Axes_3_Travel>
    <Axes_3_Angular>double</Axes_3_Angular>
    <Axes_3_Travel_speed>double</Axes_3_Travel_speed>
    <Axes_3_Rotation_Speed>double</Axes_3_Rotation_Speed>
    <Axes_4_Axes_Name>string</Axes_4_Axes_Name>
    <Axes_4_Dependence>string</Axes_4_Dependence>
    <Axes_4_Motion_type>string</Axes_4_Motion_type>
    <Axes_4_Motion_axis>string</Axes_4_Motion_axis>
    <Axes_4_Travel>double</Axes_4_Travel>
    <Axes_4_Angular>double</Axes_4_Angular>
    <Axes_4_Travel_speed>double</Axes_4_Travel_speed>
    <Axes_4_Rotation_Speed>double</Axes_4_Rotation_Speed>
    <Axes_5_Axes_Name>string</Axes_5_Axes_Name>
    <Axes_5_Dependence>string</Axes_5_Dependence>
    <Axes_5_Motion_type>string</Axes_5_Motion_type>
```

```xml
<Axes_5_Motion_axis>string</Axes_5_Motion_axis>
<Axes_5_Travel>double</Axes_5_Travel>
<Axes_5_Angular>double</Axes_5_Angular>
<Axes_5_Travel_speed>double</Axes_5_Travel_speed>
<Axes_6_Axes_Name>string</Axes_6_Axes_Name>
<Axes_6_Dependence>string</Axes_6_Dependence>
<Axes_6_Motion_type>string</Axes_6_Motion_type>
<Axes_6_Motion_axis>string</Axes_6_Motion_axis>
<Axes_6_Travel>double</Axes_6_Travel>
<Axes_6_Angular>double</Axes_6_Angular>
<Axes_6_Travel_speed>double</Axes_6_Travel_speed>
<Axes_6_Rotation_Speed>double</Axes_6_Rotation_Speed>
<Axes_7_Axes_Name>string</Axes_7_Axes_Name>
<Axes_7_Dependence>string</Axes_7_Dependence>
<Axes_7_Motion_type>string</Axes_7_Motion_type>
<Axes_7_Motion_axis>string</Axes_7_Motion_axis>
<Axes_7_Travel>double</Axes_7_Travel>
<Axes_7_Travel_speed>double</Axes_7_Travel_speed>
<Axes_8_Axes_Name>string</Axes_8_Axes_Name>
<Axes_8_Dependence>string</Axes_8_Dependence>
<Axes_8_Motion_type>string</Axes_8_Motion_type>
<Axes_8_Motion_axis>string</Axes_8_Motion_axis>
<Axes_8_Travel>double</Axes_8_Travel>
<Axes_8_Travel_speed>double</Axes_8_Travel_speed>
<Axes_9_Axes_Name>string</Axes_9_Axes_Name>
<Axes_9_Dependence>string</Axes_9_Dependence>
<Axes_9_Motion_type>string</Axes_9_Motion_type>
<Axes_9_Motion_axis>string</Axes_9_Motion_axis>
<Axes_9_Travel>double</Axes_9_Travel>
<Axes_9_Travel_speed>double</Axes_9_Travel_speed>
<Axes_10_Axes_Name>string</Axes_10_Axes_Name>
<Axes_10_Dependence>string</Axes_10_Dependence>
<Axes_10_Motion_type>string</Axes_10_Motion_type>
<Axes_10_Motion_axis>string</Axes_10_Motion_axis>
<Axes_10_Travel>double</Axes_10_Travel>
<Axes_10_Travel_speed>double</Axes_10_Travel_speed>
<Motion_Repeatability>double</Motion_Repeatability>
<Motion_Accuracy>double</Motion_Accuracy>
<Motion_Resolution>double</Motion_Resolution>
<Load_X___Axis>double</Load_X___Axis>
<Load_Y___Axis>double</Load_Y___Axis>
<Load_Z___Axis>double</Load_Z___Axis>
<_Spindle_speed>int</_Spindle_speed>
<Spindle_speed_Min>int</Spindle_speed_Min>
<Spindle_speed_Max>int</Spindle_speed_Max>
<Power>double</Power>
<Number_of_Tools>int</Number_of_Tools>
<T1_>string</T1_>
<T2>string</T2>
<T3>string</T3>
<T4>string</T4>
<Module_Name>string</Module_Name>
<Tool_Change_Time>double</Tool_Change_Time>
<Maintenance_Info>string</Maintenance_Info>
<Name__Serial_Number_>string</Name__Serial_Number_>
<Vendor_Info>string</Vendor_Info>
<Operator_Info>string</Operator_Info>
```

```
        <Manufacture_Date>string</Manufacture_Date>
        <Lubricant_>string</Lubricant_>
        <Coolant>string</Coolant>
        <Billing_rate>double</Billing_rate>
        <Purchase_price>double</Purchase_price>
        <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
        <length_of_the_loan>double</length_of_the_loan>
        <Annual_maintenance_costs>double</Annual_maintenance_costs>
        <Price_of_electricity>double</Price_of_electricity>
        <Projected_machine_use>double</Projected_machine_use>
        <cost_of_consumables>double</cost_of_consumables>
        <Depreciation_Period>double</Depreciation_Period>
        <Max_tool_length>string</Max_tool_length>
        <Control>string</Control>
        <Company>string</Company>
      </md>
    </setMachineSpecification>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setMachineSpecificationResponse xmlns="http://ifab.boeing.com/">
      <setMachineSpecificationResult>boolean</setMachineSpecificationResult>
    </setMachineSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.2.2 Add tooling T to the library

Inputs: Tooling specification

Output: Boolean

Description: Add the tooling.

API: setToolingSpecification (<tooling specification>)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setToolingSpecification xmlns="http://ifab.boeing.com/">
      <tool>
        <Id>int</Id>
        <Name>string</Name>
        <Process_1>string</Process_1>
        <Process_2>string</Process_2>
        <Process_3>string</Process_3>
        <Process_4>string</Process_4>
        <Process_5>string</Process_5>
        <Process_6>string</Process_6>
        <Process_7>string</Process_7>
        <Process_8>string</Process_8>
        <Material_1>string</Material_1>
        <Material_2>string</Material_2>
        <Material_3>string</Material_3>
        <Material_4>string</Material_4>
        <Material_5>string</Material_5>
        <Material_6>string</Material_6>
        <Material_7>string</Material_7>
        <Material_8>string</Material_8>
        <Material_1_Optimal_feed__SFM_>string</Material_1_Optimal_feed__SFM_>
        <Material_1_Optimal_speed>string</Material_1_Optimal_speed>
        <Material_1_Chipload_per_flute>double</Material_1_Chipload_per_flute>
        <Material_1_Coolant>string</Material_1_Coolant>
        <Material_2_Optimal_feed__SFM_>string</Material_2_Optimal_feed__SFM_>
        <Material_2_Optimal_speed>string</Material_2_Optimal_speed>
        <Material_2_Chipload_per_flute>double</Material_2_Chipload_per_flute>
        <Material_2_Coolant>string</Material_2_Coolant>
        <Material_3_Optimal_feed__SFM_>string</Material_3_Optimal_feed__SFM_>
        <Material_3_Optimal_speed>string</Material_3_Optimal_speed>
        <Material_3_Chipload_per_flute>double</Material_3_Chipload_per_flute>
        <Material_3_Coolant>string</Material_3_Coolant>
        <Material_4_Optimal_feed__SFM_>string</Material_4_Optimal_feed__SFM_>
        <Material_4_Optimal_speed>string</Material_4_Optimal_speed>
        <Material_4_Chipload_per_flute>double</Material_4_Chipload_per_flute>
        <Material_4_Coolant>string</Material_4_Coolant>
        <Material_5_Optimal_feed__SFM_>string</Material_5_Optimal_feed__SFM_>
        <Material_5_Optimal_speed>string</Material_5_Optimal_speed>
        <Material_5_Chipload_per_flute>double</Material_5_Chipload_per_flute>
        <Material_5_Coolant>string</Material_5_Coolant>
        <Material_6_Optimal_feed__SFM_>string</Material_6_Optimal_feed__SFM_>
        <Material_6_Optimal_speed>string</Material_6_Optimal_speed>
        <Material_6_Chipload_per_flute>double</Material_6_Chipload_per_flute>
        <Material_6_Coolant>string</Material_6_Coolant>
        <Material_7_Optimal_feed__SFM_>string</Material_7_Optimal_feed__SFM_>
        <Material_7_Optimal_speed>string</Material_7_Optimal_speed>
        <Material_7_Chipload_per_flute>double</Material_7_Chipload_per_flute>
        <Material_7_Coolant>string</Material_7_Coolant>
        <Material_8_Optimal_feed__SFM_>string</Material_8_Optimal_feed__SFM_>
        <Material_8_Optimal_speed>string</Material_8_Optimal_speed>
        <Material_8_Chipload_per_flute>double</Material_8_Chipload_per_flute>
        <Material_8_Coolant>string</Material_8_Coolant>
```

```
            <Tool_details>string</Tool_details>
            <Tool_geometry>string</Tool_geometry>
            <Overall_Length__Inches_>double</Overall_Length__Inches_>
            <Shank_diameter__Inches_>double</Shank_diameter__Inches_>
            <No__of_flutes>double</No__of_flutes>
            <Helix_angle__degrees_>double</Helix_angle__degrees_>
            <Corner_radius>string</Corner_radius>
            <Reach_length>double</Reach_length>
            <Material>string</Material>
            <Tool_material>string</Tool_material>
            <Coat_material>string</Coat_material>
            <Tool_hardness__Vickers_>double</Tool_hardness__Vickers_>
            <Optimum_tool_life>string</Optimum_tool_life>
            <Tolerance_Model>double</Tolerance_Model>
            <Management_info>string</Management_info>
            <Ordering_number>double</Ordering_number>
            <Vendor_info>string</Vendor_info>
            <Serial_number>string</Serial_number>
            <Manufacturing_date>string</Manufacturing_date>
            <Economic_info>string</Economic_info>
            <Price_of_tool__USD_>double</Price_of_tool__USD_>
            <Tool_usage__hrs_>double</Tool_usage__hrs_>
            <Maintenance_costs__USD_>double</Maintenance_costs__USD_>
            <GUID>guid</GUID>
          </tool>
      </setToolingSpecification>
    </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setToolingSpecificationResponse xmlns="http://ifab.boeing.com/">
      <setToolingSpecificationResult>boolean</setToolingSpecificationResult>
    </setToolingSpecificationResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.2.3   Add tool T to the library

Inputs: (Hand) Tool specification

Output: Boolean

Description: Add the (hand) tool.

API: setHandToolSpecification (<tool specification>)

Status: Limited availability

Approved for public release; distribution unlimited.

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setHandToolSpecification xmlns="http://ifab.boeing.com/">
      <specs>
        <Id>int</Id>
        <Name>string</Name>
        <Hand_tool>string</Hand_tool>
        <Process_1>string</Process_1>
        <Mounting___Spline>string</Mounting___Spline>
        <Weight_excluding_chuck_collet>double</Weight_excluding_chuck_collet>
        <Weight_including_chuck_collet>double</Weight_including_chuck_collet>
        <Chuck_collet_size>string</Chuck_collet_size>
        <Air_hose_inlet_size>double</Air_hose_inlet_size>
        <Min_socket>string</Min_socket>
        <Max_socket>string</Max_socket>
        <Cutting_pad_diameter>double</Cutting_pad_diameter>
        <Alumina_nozzle>double</Alumina_nozzle>
        <AC_Voltage>double</AC_Voltage>
        <AC_Current>double</AC_Current>
        <Battery_Type>string</Battery_Type>
        <Battery_Voltage>string</Battery_Voltage>
        <Battery_Capacity>string</Battery_Capacity>

<Air_Consumption_rate_at_max_output>double</Air_Consumption_rate_at_max_output>

<Air_Consumption_rate_at_free_speed>double</Air_Consumption_rate_at_free_speed>

<Pnuematic_recommended_hose_size>double</Pnuematic_recommended_hose_size>
        <Operating_temperature_min>double</Operating_temperature_min>
        <Operating_temperature_max>double</Operating_temperature_max>
        <Sound_level>double</Sound_level>
        <Max_power_output>double</Max_power_output>
        <Max_operating_pressure>double</Max_operating_pressure>
        <Measured_vibration_value>double</Measured_vibration_value>
        <Max_cutting_depth>string</Max_cutting_depth>
        <Setting_range>double</Setting_range>
        <Duty_Cycle>double</Duty_Cycle>
        <Rated_Amps>double</Rated_Amps>
        <Input_Power>string</Input_Power>
        <Input_Current_at_Rated_Output>string</Input_Current_at_Rated_Output>
        <Rated_Output>string</Rated_Output>
        <Torque_Min>double</Torque_Min>
        <Torque_Max>double</Torque_Max>
        <Max_Torque>string</Max_Torque>
```

```xml
            <Speed_at_max_output>string</Speed_at_max_output>
            <Speed_at_free_speed>double</Speed_at_free_speed>
            <Impacts_min>string</Impacts_min>
            <Gear_ratio>string</Gear_ratio>
            <Length_feed>string</Length_feed>
            <Stroke>string</Stroke>
            <CS_distance>double</CS_distance>
            <Tool_Envelope_Length>double</Tool_Envelope_Length>
            <Tool_Envelope_Width>double</Tool_Envelope_Width>
            <Tool_Envelope_Height>double</Tool_Envelope_Height>
            <Height_over_spindle_D>string</Height_over_spindle_D>
            <Height_over_spindle_T>string</Height_over_spindle_T>
            <Height_over_spindle_H>double</Height_over_spindle_H>
            <Angle_head_height_>double</Angle_head_height_>
            <Chuck_Collet_T1>string</Chuck_Collet_T1>
            <Chuck_Collet_T2>string</Chuck_Collet_T2>
            <Hook_up_Kit_Length>double</Hook_up_Kit_Length>
            <Hook_Up_Kit_Width_>double</Hook_Up_Kit_Width_>
            <Hook_up_Kit_Height>double</Hook_up_Kit_Height>
            <Hook_up_Kit_Weight>double</Hook_up_Kit_Weight>
            <Drive_Square_>string</Drive_Square_>
            <Drive_Hexa>string</Drive_Hexa>
            <Drive_Ratchet>string</Drive_Ratchet>
            <Model_number>string</Model_number>
            <Hand_tool1>string</Hand_tool1>
            <Abrasive_wheel>string</Abrasive_wheel>
            <Service_kit>string</Service_kit>
            <Vendor_info>string</Vendor_info>
            <Manufacturing_date>dateTime</Manufacturing_date>
            <CAD_Drawing_file>string</CAD_Drawing_file>
            <Billing_rate>double</Billing_rate>
            <Life_time_maintenance_costs>double</Life_time_maintenance_costs>
            <Annual_maintenance_costs>double</Annual_maintenance_costs>
            <Price_of_electricity___KW>double</Price_of_electricity___KW>

<Projected_machine_hours___year>double</Projected_machine_hours___year>
            <cost_of_consumables>double</cost_of_consumables>
            <Price>double</Price>
          </specs>
        </setHandToolSpecification>
      </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <setHandToolSpecificationResponse xmlns="http://ifab.boeing.com/">

<setHandToolSpecificationResult>boolean</setHandToolSpecificationResult>
    </setHandToolSpecificationResponse>
  </soap12:Body>
```

Approved for public release; distribution unlimited.

```
</soap12:Envelope>
```

1.2.4   Get Machine IDs

Inputs:

Output: List of <machine name, machine ID> pairs

Description: List the machine IDs.

API: getMachineNamesIDs ()

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNamesIDs xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getMachineNamesIDsResponse xmlns="http://ifab.boeing.com/">
      <getMachineNamesIDsResult>
        <string>string</string>
        <string>string</string>
      </getMachineNamesIDsResult>
    </getMachineNamesIDsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.2.5   Get Tooling IDs

Inputs:

Output: List of <tooling name, tooling ID> pairs

Description: List the tooling IDs.

Approved for public release; distribution unlimited.

API: getToolingNamesIDs ()

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNamesIDs xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getToolingNamesIDsResponse xmlns="http://ifab.boeing.com/">
      <getToolingNamesIDsResult>
        <string>string</string>
        <string>string</string>
      </getToolingNamesIDsResult>
    </getToolingNamesIDsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.2.6   Get Tool IDs

Inputs:

Output: List of <tool name, tool ID> pairs

Description: List the tool IDs.

API: getHandToolNamesIDs ()

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
```

Approved for public release; distribution unlimited.

```
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNamesIDs xmlns="http://ifab.boeing.com/" />
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getHandToolNamesIDsResponse xmlns="http://ifab.boeing.com/">
      <getHandToolNamesIDsResult>
        <string>string</string>
        <string>string</string>
      </getHandToolNamesIDsResult>
    </getHandToolNamesIDsResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.2.7   Delete machine M from the library

Inputs: Machine ID

Output: Boolean

Description: Delete the machine.

API: deleteMachine (string machineID)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteMachine xmlns="http://ifab.boeing.com/">
      <ID>int</ID>
    </deleteMachine>
```

Approved for public release; distribution unlimited.

```
    </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteMachineResponse xmlns="http://ifab.boeing.com/">
      <deleteMachineResult>boolean</deleteMachineResult>
    </deleteMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.2.8 Delete tooling T from the library

Inputs: Tooling ID

Output: Boolean

Description: Delete the tooling.

API: deleteTooling (string toolingID)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteTooling xmlns="http://ifab.boeing.com/">
      <Id>int</Id>
    </deleteTooling>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
```

Approved for public release; distribution unlimited.

```
      <deleteToolingResponse xmlns="http://ifab.boeing.com/">
        <deleteToolingResult>boolean</deleteToolingResult>
      </deleteToolingResponse>
    </soap12:Body>
</soap12:Envelope>
```

1.2.9   Delete tool T from the library

Inputs: (Hand) Tool ID

Output: Boolean

Description: Delete the (hand) tool.

API: deleteHandTool (string toolID)

Status: Limited availability

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteHandTool xmlns="http://ifab.boeing.com/">
      <Id>int</Id>
    </deleteHandTool>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <deleteHandToolResponse xmlns="http://ifab.boeing.com/">
      <deleteHandToolResult>boolean</deleteHandToolResult>
    </deleteHandToolResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.3 DESIGN SPECIFIC FABRICATION OPERATIONS

These operations are design specific in that they have one or more inputs that depend upon the characteristics of a design, rather than being answerable solely based on information in the library.

1.3.1   Can I fit (rectangular) workpiece W on machine M

Inputs: Workpiece

Machine name

Output: Boolean

Description: This query is for non-turning machines. Does the workpiece fit on the table for the machine, is the weight of the workpiece less than the weight the table can handle. The workpiece is represented by a bounding box, weight and material, and the caller will extract that information from the META TDB or some other source.

API: fitsOnMachine(string machineName, float length, float width, float height, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fitsOnMachine xmlns="http://ifab.boeing.com/">
      <machineName>string</machineName>
      <x>float</x>
      <y>float</y>
      <z>float</z>
      <weight>float</weight>
      <materialName>string</materialName>
    </fitsOnMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```
  <soap12:Body>
    <fitsOnMachineResponse xmlns="http://ifab.boeing.com/">
      <fitsOnMachineResult>boolean</fitsOnMachineResult>
    </fitsOnMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.3.2   Can I fit (cylindrical) workpiece W on machine M

Inputs: Workpiece

Machine name

Output: Boolean

Description: This query is for turning machines. Does the workpiece fit on the table for the machine, is the weight of the workpiece less than the weight the table can handle. The workpiece is represented by a length and diameter, weight and material, and the caller will extract that information from the META TDB or some other source.

API: fitsOnMachine2 (string machineName, float Z, float diameter, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <fitsOnMachine2 xmlns="http://ifab.boeing.com/">
      <machineName>string</machineName>
      <z>float</z>
      <diameter>float</diameter>
      <weight>float</weight>
      <materialName>string</materialName>
    </fitsOnMachine2>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

Approved for public release; distribution unlimited.

```
  <soap12:Body>
    <fitsOnMachine2Response xmlns="http://ifab.boeing.com/">
      <fitsOnMachine2Result>boolean</fitsOnMachine2Result>
    </fitsOnMachine2Response>
  </soap12:Body>
</soap12:Envelope>
```

1.3.3   What machines can perform process P

Inputs: process

Output: list of machines names

Description: List of machines that can perform the required process

API: capableMachines (string processName)

Status: Available

Type: Pure data

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachines xmlns="http://MCPML_URI/">
      <processName>string</processName>
    </capableMachines>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachinesResponse xmlns="http://MCPML_URI/">
      <capableMachinesResult>
        <string>string</string>
        <string>string</string>
      </capableMachinesResult>
    </capableMachinesResponse>
  </soap12:Body>
</soap12:Envelope>
```

Approved for public release; distribution unlimited.

1.3.4 What machines can perform process P on (rectangular) workpiece W

Inputs: process

Workpiece

Output: list of machine names

Description: For rectangular workpieces on non-turning machines. List of machines that can perform process P and have a worktable that can accommodate the workpiece. Workpiece is defined by its bounding box, weight, and material.

API: capableMachinesWorkpiece (string processName, float length, float width, float height, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachinesWorkpiece xmlns="http://MCPML_URI/">
      <processName>string</processName>
      <length>int</length>
      <width>int</width>
      <height>int</height>
      <weight>int</weight>
      <materialName>string</materialName>
    </capableMachinesWorkpiece>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachinesWorkpieceResponse xmlns="http://MCPML_URI/">
      <capableMachinesWorkpieceResult>
        <string>string</string>
        <string>string</string>
      </capableMachinesWorkpieceResult>
```

Approved for public release; distribution unlimited.

```
      </capableMachinesWorkpieceResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.3.5   What machines can perform process P on (cylindrical) workpiece W

Inputs: process

Workpiece

Output: list of machine names

Description: For cylindrical workpieces on turning machines. List of machines that can perform process P and have a worktable that can accommodate the workpiece. Workpiece is defined by its bounding box, weight, and material.

 API: capableMachinesWorkpiece2 (string processName, float Z, float diameter, float weight, string materialName)

Status: Available

Type: Business logic

```
API: capableMachinesWorkpiece2 (string processName, float Z, float diameter,
float weight, string materialName) POST /mcpml/iFAB_Service.asmx HTTP/1.1
```
```
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <capableMachinesWorkpiece2 xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <z>float</z>
      <diameter>float</diameter>
      <weight>float</weight>
      <materialName>string</materialName>
    </capableMachinesWorkpiece2>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
```

Approved for public release; distribution unlimited.

```
    <capableMachinesWorkpiece2Response xmlns="http://ifab.boeing.com/">
      <capableMachinesWorkpiece2Result>
        <string>string</string>
        <string>string</string>
      </capableMachinesWorkpiece2Result>
    </capableMachinesWorkpiece2Response>
  </soap12:Body>
</soap12:Envelope>
```

Status: Available

1.3.6 What tools can be used to perform process P on workpiece W on machine M

Inputs: Machine name

Process name

Workpiece

Output: list of tools

Description: List of tools that can be used by machine M to perform non-turning process P on rectangular workpiece W. The workpiece is defined by its bounding box, weight, and material. List of tools is empty if M doesn't support P, or W fit on M.

API: toolsForProcessOnMachine (string processName, float length, float width, float height, float weight, string materialName, string machineName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForProcessOnMachine xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <length>float</length>
      <width>float</width>
      <height>float</height>
      <weight>float</weight>
```

Approved for public release; distribution unlimited.

```
        <materialName>string</materialName>
        <machineName>string</machineName>
    </toolsForProcessOnMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForProcessOnMachineResponse xmlns="http://ifab.boeing.com/">
      <toolsForProcessOnMachineResult>
        <string>string</string>
        <string>string</string>
      </toolsForProcessOnMachineResult>
    </toolsForProcessOnMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.3.7　What tools can be used to perform turning process P on workpiece W on machine M

Inputs: Machine name

Process name

Workpiece

Output: list of tools

Description: List of tools that can be used by machine M to perform turning process P on cylindrical workpiece W. The workpiece is defined by its diameter, length, weight, and material. List of tools is empty if process is not a turning process, M doesn't support P, or W fit on M.

API: toolsForTurningProcessOnMachine (string processName, float diameter, float length, float weight, string materialName, string machineName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

Approved for public release; distribution unlimited.

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForTurningProcessOnMachine xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <diameter>float</diameter>
      <length>float</length>
      <weight>float</weight>
      <materialName>string</materialName>
      <machineName>string</machineName>
    </toolsForTurningProcessOnMachine>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <toolsForTurningProcessOnMachineResponse xmlns="http://ifab.boeing.com/">
      <toolsForTurningProcessOnMachineResult>
        <string>string</string>
        <string>string</string>
      </toolsForTurningProcessOnMachineResult>
    </toolsForTurningProcessOnMachineResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.3.8   What processes can be used to add feature F to workpiece W

Inputs: Feature name

Workpiece

Output: List of process names

Description: Return a list of processes that can be used to create feature F on a workpiece. Processes provide information about the kinds of features they can create. They may also be constrained to only apply to workpieces satisfying certain requirements (e.g., material, thickness), and the workpiece information is provided to support evaluating those constraints.

We assume that the feature is the result of a single operation on a single machine, so sequencing is not addressed by this query.

API: relevantProcessForFeatureWorkpiece (string featureName, float length, float width, float height, float weight, string materialName)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <relevantProcessForFeatureWorkpiece xmlns="http://ifab.boeing.com/">
      <featureName>string</featureName>
      <length>float</length>
      <width>float</width>
      <height>float</height>
      <weight>float</weight>
      <materialName>string</materialName>
    </relevantProcessForFeatureWorkpiece>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <relevantProcessForFeatureWorkpieceResponse
xmlns="http://ifab.boeing.com/">
      <relevantProcessForFeatureWorkpieceResult>
        <string>string</string>
        <string>string</string>
      </relevantProcessForFeatureWorkpieceResult>
    </relevantProcessForFeatureWorkpieceResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.3.9 What machines and tools can be used to add feature F to workpiece W using process P

Inputs: process

Feature name

Workpiece (size, weight, material)

Output:

List of <machine, tool> pairs

Description:List of machines and tools that can be used to create feature F in workpiece W using process P. Workpiece is characterized by its bounding box, weight, and material. Feature is characterized by its type.

API: machineToolsForFeature (string processName, sting featureName, float workpiecelength, float workpiecewidth, float workpieceheight, float workpieceweight, string maerialName)

Status: In Development

Type: Business logic

1.3.10 What resources are required to add feature F to workpiece W using process P with Machine M and tool T

Inputs: process

Feature

Workpiece

Output: list of resources

Description: Includes the non-machine/tool resources required (operators, fixtures, cranes, infrastructure, power, etc.).

API: resourcesForFeature (string processName, float featureLength, float featureWidth, float featureHeight, ToleranceType tolerance, float workpieceLength, float workpieceWidth, float workpieceHeight, float workpieceWeight, string machineName, string toolName)

Status: In Development

Type: Business logic

1.3.11 How long does it take to apply feature F to workpiece W using process P on machine M using tool T

Inputs: feature

Workpiece

Process

Machine

Tool

Output: time in seconds or n/a

Description: Time required to apply feature to workpiece. Assumes a single tool and a single operation. Sequencing is performed at a higher level.

API:

Status: In Development

Type: Business logic

1.3.12 How much does it cost to apply feature F to workpiece W using process P on machine M using tool T

Inputs: feature

Workpiece

Process

Machine

Tool

Output: cost in dollars or n/a

Description: Cost to apply feature, primarily due to machine and tool usage and tool wear.

API:

Status: In Development

Type: Business logic

1.3.13 What tolerance is achieved when applying feature F to workpiece W using process P on machine M using tool T

Inputs: feature

Workpiece

Process

Machine

Tool

Output: tolerance or n/a

Description: Tolerance achieved when using the specified resources to produce the desired feature.

API:

Status: In Development

Type: Business logic

1.3.14 How long does it take to apply feature F to workpiece W using process P on machine M using tool T

Inputs: Feature (name, length, width, height)

Workpiece (length, width, height, weight, material)

Process

Machine

Tooling

Tolerance (TBD)

Output: time in seconds or n/a

Description: Time required to apply feature to workpiece. Assumes a single tool and a single operation. Sequencing is performed at a higher level.

API: timeForFeature(string processName, string featureName, float featureLength, float featureWidth, float featureHeight, float workpieceLength, float workpieceWidth, float workpieceHeight, float workpieceWeight, string machineName, string toolingName, string materialName)

Status: In Development

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForFeature xmlns="http://ifab.boeing.com/">
      <processName>string</processName>
      <featureName>string</featureName>
      <featureLength>float</featureLength>
      <featureWidth>float</featureWidth>
      <featureHeight>float</featureHeight>
      <tolerance />
      <workpieceLength>float</workpieceLength>
      <workpieceWidth>float</workpieceWidth>
      <workpieceHeight>float</workpieceHeight>
      <workpieceWeight>float</workpieceWeight>
```

```
      <machineName>string</machineName>
      <toolName>string</toolName>
      <materialName>string</materialName>
    </timeForFeature>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <timeForFeatureResponse xmlns="http://ifab.boeing.com/">
      <timeForFeatureResult>float</timeForFeatureResult>
    </timeForFeatureResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.4 DESIGN SPECIFIC ASSEMBLY OPERATIONS

These operations are design specific in that they have one or more inputs that depend upon the characteristics of a design, rather than being answerable solely based on information in the library.

1.4.1   What processes can I use to assemble part X to Y using fastening method F

Inputs: Part X (size, weight)

Part Y (size, weight)

Fastening method (type, mechanism)

Output: Process

Description: Processes that can be used to assemble Part X to Part Y using the selected fastening method. The parts are described by their bounding boxes and weights.

API: getCandidateFastenProcesses (float part1X, part1Y, part1Z, part1Weight, part2X, part2Y, part2Z, part2Weight, string fasteningMethod, string fasteningMedium)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

Approved for public release; distribution unlimited.

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCandidateFastenProcesses xmlns="http://MCPML_URI/">
      <part1X>float</part1X>
      <part1Y>float</part1Y>
      <part1Z>float</part1Z>
      <part1weight>float</part1weight>
      <part2X>float</part2X>
      <part2Y>float</part2Y>
      <part2Z>float</part2Z>
      <part2weight>float</part2weight>
      <FasteningMethod>string</FasteningMethod>
      <FasteningMedium>string</FasteningMedium>
    </getCandidateFastenProcesses>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getCandidateFastenProcessesResponse xmlns="http://MCPML_URI/">
      <getCandidateFastenProcessesResult>
        <executeTime_minutes>string</executeTime_minutes>
        <parts>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
        </parts>
        <fasteners>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
```

```
          </fasteners>
          <resources>
            <string>string</string>
            <string>string</string>
          </resources>
          <sequence>
            <ProcessActivity>
              <name>string</name>
              <activityId>int</activityId>
              <taskSets xsi:nil="true" />
            </ProcessActivity>
            <ProcessActivity>
              <name>string</name>
              <activityId>int</activityId>
              <taskSets xsi:nil="true" />
            </ProcessActivity>
          </sequence>
        </getCandidateFastenProcessesResult>
      </getCandidateFastenProcessesResponse>
    </soap12:Body>
</soap12:Envelope>
```

1.4.2   What is the time for assembling part X to part Y using fastening method F and process P

Inputs: Part X (size, weight)

Part Y (size, weight)

Fastening method (type, mechanism)

Process

Outputs:

Assembly time in minutes

Description: Time required to assemble Part X to Part Y using the selected fastening method. The parts are described by their bounding boxes and weights.

API: getAssemblyTimeForFastenProcess (float part1X, float part1Y, float part1Z, float part1weight, float part2X, float part2Y, float part2Z, float part2weight, string FasteningMethod, string FasteningMedium, int FastenerCount, string process, string resources)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```

Approved for public release; distribution unlimited.

```xml
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForFastenProcess xmlns="http://MCPML_URI/">
      <part1X>float</part1X>
      <part1Y>float</part1Y>
      <part1Z>float</part1Z>
      <part1weight>float</part1weight>
      <part2X>float</part2X>
      <part2Y>float</part2Y>
      <part2Z>float</part2Z>
      <part2weight>float</part2weight>
      <FasteningMethod>string</FasteningMethod>
      <FasteningMedium>string</FasteningMedium>
      <FastenerCount>int</FastenerCount>
      <Process>string</Process>
      <Resources>string</Resources>
    </getAssemblyTimeForFastenProcess>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForFastenProcessResponse xmlns="http://MCPML_URI/">
      <getAssemblyTimeForFastenProcessResult>
        <executeTime_minutes>string</executeTime_minutes>
        <parts>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
        </parts>
        <fasteners>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
```

Approved for public release; distribution unlimited.

```
            <Count>int</Count>
          </Fastener>
        </fasteners>
        <resources>
          <string>string</string>
          <string>string</string>
        </resources>
        <sequence>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
        </sequence>
      </getAssemblyTimeForFastenProcessResult>
    </getAssemblyTimeForFastenProcessResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.4.3   How long does it take to assemble part X to part Y using tool T

Inputs: Part X (size, weight)

Part Y (size, weight)

Tool (name)

Output: Time in minutes

Description: : Time required to assemble Part X to Part Y using the selected tool. The parts are described by their bounding boxes and weights.

API: getAssemblyTimeForFastenTool (float part1X, float part1Y, float part1Z, float part1weight, float part2X, float part2Y, float part2Z, float part2weight, string toolName, string resources)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

Approved for public release; distribution unlimited.

```
   <soap12:Body>
     <getAssemblyTimeForFastenTool xmlns="http://ifab.boeing.com/">
       <part1X>float</part1X>
       <part1Y>float</part1Y>
       <part1Z>float</part1Z>
       <part1weight>float</part1weight>
       <part2X>float</part2X>
       <part2Y>float</part2Y>
       <part2Z>float</part2Z>
       <part2weight>float</part2weight>
       <tool>string</tool>
       <resources>string</resources>
     </getAssemblyTimeForFastenTool>
   </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
   <soap12:Body>
     <getAssemblyTimeForFastenToolResponse xmlns="http://ifab.boeing.com/">
       <getAssemblyTimeForFastenToolResult>
         <executeTime_minutes>string</executeTime_minutes>
         <parts>
           <Part>
             <xDimension_in>float</xDimension_in>
             <yDimension_in>float</yDimension_in>
             <zDimension_in>float</zDimension_in>
             <weight_in_lb>float</weight_in_lb>
           </Part>
           <Part>
             <xDimension_in>float</xDimension_in>
             <yDimension_in>float</yDimension_in>
             <zDimension_in>float</zDimension_in>
             <weight_in_lb>float</weight_in_lb>
           </Part>
         </parts>
         <fasteners>
           <Fastener>
             <Method>string</Method>
             <Medium>string</Medium>
             <Count>int</Count>
           </Fastener>
           <Fastener>
             <Method>string</Method>
             <Medium>string</Medium>
             <Count>int</Count>
           </Fastener>
         </fasteners>
         <resources>
           <string>string</string>
           <string>string</string>
```

```
      </resources>
      <sequence>
        <ProcessActivity>
          <name>string</name>
          <activityId>int</activityId>
          <taskSets xsi:nil="true" />
        </ProcessActivity>
        <ProcessActivity>
          <name>string</name>
          <activityId>int</activityId>
          <taskSets xsi:nil="true" />
        </ProcessActivity>
      </sequence>
    </getAssemblyTimeForFastenToolResult>
  </getAssemblyTimeForFastenToolResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.4.4   How long does it take to assemble part X to part Y using fastening method F

Inputs: Part X (size, weight)

Part Y (size, weight)

Fastening method (type, mechanism)

Output: Time in minutes

Description: : Time required to assemble Part X to Part Y using the selected fastening method. The parts are described by their bounding boxes and weights.

API: getAssemblyFastenTime (float part1X, float part1Y, float part1Z, float part1weight, float part2X, float part2Y, float part2Z, float part2weight, string FasteningMethod, string FasteningMedium, int FastenerCount)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyFastenTime xmlns="http://MCPML_URI/">
      <part1X>float</part1X>
      <part1Y>float</part1Y>
```

Approved for public release; distribution unlimited.

```
      <part1Z>float</part1Z>
      <part1weight>float</part1weight>
      <part2X>float</part2X>
      <part2Y>float</part2Y>
      <part2Z>float</part2Z>
      <part2weight>float</part2weight>
      <FasteningMethod>string</FasteningMethod>
      <FasteningMedium>string</FasteningMedium>
      <FastenerCount>int</FastenerCount>
    </getAssemblyFastenTime>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyFastenTimeResponse xmlns="http://MCPML_URI/">
      <getAssemblyFastenTimeResult>
        <executeTime_minutes>string</executeTime_minutes>
        <parts>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
        </parts>
        <fasteners>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
        </fasteners>
        <resources>
          <string>string</string>
          <string>string</string>
        </resources>
        <sequence>
          <ProcessActivity>
            <name>string</name>
```

Approved for public release; distribution unlimited.

```
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
        </sequence>
      </getAssemblyFastenTimeResult>
    </getAssemblyFastenTimeResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.4.5   What is the time for inserting part X into part Y

Inputs: Part X (size, weight)

Part Y (size, weight)

Resources

Outputs:

Assembly time in minutes

Description: Time required to insert Part X into Part Y. The parts are described by their bounding boxes and weights.

API: getAssemblyTimeForInsert (float part1X, float part1Y, float part1Z, float part1weight, float part2X, float part2Y, float part2Z, float part2weight, string resources)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForInsert xmlns="http://ifab.boeing.com/">
      <part1X>float</part1X>
      <part1Y>float</part1Y>
      <part1Z>float</part1Z>
      <part1weight>float</part1weight>
      <part2X>float</part2X>
      <part2Y>float</part2Y>
      <part2Z>float</part2Z>
```

Approved for public release; distribution unlimited.

```
      <part2weight>float</part2weight>
      <resources>string</resources>
    </getAssemblyTimeForInsert>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForInsertResponse xmlns="http://ifab.boeing.com/">
      <getAssemblyTimeForInsertResult>
        <executeTime_minutes>string</executeTime_minutes>
        <parts>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
        </parts>
        <fasteners>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
        </fasteners>
        <resources>
          <string>string</string>
          <string>string</string>
        </resources>
        <sequence>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
```

Approved for public release; distribution unlimited.

```
          <taskSets xsi:nil="true" />
        </ProcessActivity>
      </sequence>
    </getAssemblyTimeForInsertResult>
  </getAssemblyTimeForInsertResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.4.6   What is the time for loose assembling part X and part Y

Inputs: Part X (size, weight)

Part Y (size, weight)

Resources

Outputs:

Assembly time in minutes

Description: Time required to loose assemble Part X to Part Y. The parts are described by their bounding boxes and weights.

API: getAssemblyTimeForFastenProcess (float part1X, float part1Y, float part1Z, float part1weight, float part2X, float part2Y, float part2Z, float part2weight, string resources)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForLooseAssemble xmlns="http://ifab.boeing.com/">
      <part1X>float</part1X>
      <part1Y>float</part1Y>
      <part1Z>float</part1Z>
      <part1weight>float</part1weight>
      <part2X>float</part2X>
      <part2Y>float</part2Y>
      <part2Z>float</part2Z>
      <part2weight>float</part2weight>
      <resources>string</resources>
    </getAssemblyTimeForLooseAssemble>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
```

```
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForLooseAssembleResponse xmlns="http://ifab.boeing.com/">
      <getAssemblyTimeForLooseAssembleResult>
        <executeTime_minutes>string</executeTime_minutes>
        <parts>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
        </parts>
        <fasteners>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
        </fasteners>
        <resources>
          <string>string</string>
          <string>string</string>
        </resources>
        <sequence>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
        </sequence>
      </getAssemblyTimeForLooseAssembleResult>
    </getAssemblyTimeForLooseAssembleResponse>
  </soap12:Body>
```

```
</soap12:Envelope>
```

1.4.7 What is the time for press fitting part X to part Y

Inputs: Part X (size, weight)

Part Y (size, weight)

Resources

Outputs:

Assembly time in minutes

Description: Time required to press fit Part X to Part Y. The parts are described by their bounding boxes and weights.

API: getAssemblyTimeForPressFit (float part1X, float part1Y, float part1Z, float part1weight, float part2X, float part2Y, float part2Z, float part2weight, string resources)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForPressFit xmlns="http://ifab.boeing.com/">
      <part1X>float</part1X>
      <part1Y>float</part1Y>
      <part1Z>float</part1Z>
      <part1weight>float</part1weight>
      <part2X>float</part2X>
      <part2Y>float</part2Y>
      <part2Z>float</part2Z>
      <part2weight>float</part2weight>
      <resources>string</resources>
    </getAssemblyTimeForPressFit>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
```

```xml
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForPressFitResponse xmlns="http://ifab.boeing.com/">
      <getAssemblyTimeForPressFitResult>
        <executeTime_minutes>string</executeTime_minutes>
        <parts>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
        </parts>
        <fasteners>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
        </fasteners>
        <resources>
          <string>string</string>
          <string>string</string>
        </resources>
        <sequence>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
        </sequence>
      </getAssemblyTimeForPressFitResult>
    </getAssemblyTimeForPressFitResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.4.8 What is the time for assembling part X to part Y using process P

Inputs: Part X (size, weight)

Part Y (size, weight)

Process (AssemblyTaskSetID)

Resources

Outputs:

Assembly time in minutes

Description: Time required to assemble Part X to Part Y using the selected process, as defined by its Assembly Task Set ID (available from the getProcessSpecifications service (1.1.5)). The parts are described by their bounding boxes and weights.

API: getAssemblyTimeForProcess (float part1X, float part1Y, float part1Z, float part1weight, float part2X, float part2Y, float part2Z, float part2weight, int processActivitySetID, string resources)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: cins-sql-win2008-02.stl.mo.boeing.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForProcess xmlns="http://ifab.boeing.com/">
      <part1X>float</part1X>
      <part1Y>float</part1Y>
      <part1Z>float</part1Z>
      <part1weight>float</part1weight>
      <part2X>float</part2X>
      <part2Y>float</part2Y>
      <part2Z>float</part2Z>
      <part2weight>float</part2weight>
      <AssemblyTaskSetID>int</AssemblyTaskSetID>
      <resources>string</resources>
    </getAssemblyTimeForProcess>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getAssemblyTimeForProcessResponse xmlns="http://ifab.boeing.com/">
      <getAssemblyTimeForProcessResult>
        <executeTime_minutes>string</executeTime_minutes>
        <parts>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
          <Part>
            <xDimension_in>float</xDimension_in>
            <yDimension_in>float</yDimension_in>
            <zDimension_in>float</zDimension_in>
            <weight_in_lb>float</weight_in_lb>
          </Part>
        </parts>
        <fasteners>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
          <Fastener>
            <Method>string</Method>
            <Medium>string</Medium>
            <Count>int</Count>
          </Fastener>
        </fasteners>
        <resources>
          <string>string</string>
          <string>string</string>
        </resources>
        <sequence>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
          <ProcessActivity>
            <name>string</name>
            <activityId>int</activityId>
            <taskSets xsi:nil="true" />
          </ProcessActivity>
        </sequence>
      </getAssemblyTimeForProcessResult>
    </getAssemblyTimeForProcessResponse>
  </soap12:Body>
</soap12:Envelope>
```

1.4.9　How long does it take, and what is the manpower required, to execute Bill of Process Element P for Part X and Part Y

Inputs: BOP Element (process type)

Part X (type, category, size, weight)

Part Y (type, category, size, weight)

Output: Process, resources, time

Description: Specific BOP process used to perform specified process for the selected part X and part Y. The parts are specified by type, class, size (bounding box), and weight. The specific process includes information about required resources, and the elapsed time, staffing, and man-hours required for each BOP process step, and for the complete operation.

API: getBillOfProcess (string ProcessType, string part1Type, string part1Category, part1Length, part1Width, part1Height, part1Weight, string part2Type, string part2Category, part2Length, part2Width, part2Height, part2Weight)

Status: Available

Type: Business logic

```
POST /mcpml/iFAB_Service.asmx HTTP/1.1
Host: MCPML_HOST
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBillOfProcess xmlns="http://MCPML_URI/">
      <processType>string</processType>
      <part1Type>string</part1Type>
      <part1Cat>string</part1Cat>
      <part1Length>double</part1Length>
      <part1Width>double</part1Width>
      <part1Height>double</part1Height>
      <part1Weight>double</part1Weight>
      <part2Type>string</part2Type>
      <part2Cat>string</part2Cat>
      <part2Length>double</part2Length>
      <part2Width>double</part2Width>
      <part2Height>double</part2Height>
      <part2Weight>double</part2Weight>
    </getBillOfProcess>
  </soap12:Body>
</soap12:Envelope>
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
```

Approved for public release; distribution unlimited.

```
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <getBillOfProcessResponse xmlns="http://MCPML_URI/">
      <getBillOfProcessResult>
        <name>string</name>
        <description>string</description>
        <elapsedTime>double</elapsedTime>
        <manHours>double</manHours>
        <maxHeads>double</maxHeads>
        <taskSteps>
          <taskStep>
            <stepName>string</stepName>
            <stepDescription>string</stepDescription>
            <stepTime>double</stepTime>
            <stepHeads>double</stepHeads>
            <resourceList xsi:nil="true" />
          </taskStep>
          <taskStep>
            <stepName>string</stepName>
            <stepDescription>string</stepDescription>
            <stepTime>double</stepTime>
            <stepHeads>double</stepHeads>
            <resourceList xsi:nil="true" />
          </taskStep>
        </taskSteps>
      </getBillOfProcessResult>
    </getBillOfProcessResponse>
  </soap12:Body>
</soap12:Envelope>
```

2.0 Upcoming Services

In addition to the services above listed as in development, other candidate services for inclusion in the library interface include services that user operator certifications to constrain choice of process or machine, and assembly process queries that use CMU/UMarlyand motion planning algorithms to validate geometric feasibility of assembly steps. We will also review the add/delete operations and their CONOPS and impact on the newly added GUID fields. We are in the process of extending the library to capture information about specific foundries (numbers of instances of the various resource types, and potentially other instance data), and making corresponding extensions to the web services (e.g., adding a foundry name argument to the getMachineNames service). We are also adding library definitions and supporting queries for fixtures, including properties of existing fixtures and modular fixtures, as well as additional materials handling models and services.

3.0 Appendix 1 – Definitions

MCPML – Manufacturing Capabilities and Process Models Library – the database

TDP – Technical Data Package – the design produced by META

4.0 Appendix 2 – Interface Specification

The MCPML is accessed via web services. This section provides a list of the current web services and sample SOAP 1.2 requests and responses. The WSDL definition file is available in the distribution. The placeholders for Host and Namespace below would be replaced by the final deployment host and namespace which will be defined in the final WSDL file and associated release notes.

5.0 Appendix 3 - Overall Query Candidate List

This appendix provides a list of questions that motivate the design of the MCPML and the various interfaces defined above. The questions are grouped into three levels of abstraction. One, fairly high, represents the basic questions that users of the library need answer in the process of configuring a foundry or determining the manufacturability of a design. The next level represents more detailed questions that presumably need to be answered in the process of creating, choosing, or defining a configuration or determining the viability of a particular manufacturing plan. These are refined into the lowest level queries that the library can either directly answer, or that the library supplies information that allows the library users (manufacturability and foundry configuration algorithms) to directly answer, in the process of configuring a foundry to manufacture a product or whether or not a particular part can be fabricated. It is this lowest level of questions that is addressed by the services defined in this interface document. The higher level questions are presumably answered by library users, building upon the library supplied data and services with for example job shop scheduling and physical layout algorithms.

The questions are furthermore divided into qualitative and quantitative questions. The quantitative questions tend to address both cost and schedule, and consider per copy cost and schedule, configuration cost and schedule, and first item cost and schedule.

5.1 HIGH LEVEL QUESTIONS

5.1.1   Questions about Configurations

Can foundry X make product Y?

How can foundry X make product Y?

5.1.2   Qualitative Configuration Questions

How much will it cost for foundry X to make product Y?

How much will it cost for foundry X to make each product Y?

How much will it cost to configure foundry X to make product Y?

How much is the overhead cost for foundry X to make product Y?

How long will it take for foundry X to make each product Y?

How long will it take to configure foundry X to make product Y?

How long will it take for foundry X to make the first product Y?

At what rate can foundry X make product Y?

What is the best way for foundry X to make product Y?

What is the fastest way for foundry X to make product Y?

What is the fastest way for foundry X to make the first product Y?

What is the fastest way for foundry X to make each product Y?

What is the cheapest way for foundry X to make product Y?

What is the cheapest way for foundry X to make the first product Y?

What is the cheapest way for foundry X to make each product Y?

5.1.3   Foundry Modification Questions

How can foundry X be modified so that it can make product Y?

How can foundry X be modified so that it can make product Y in time T?

How can foundry X be modified so that it can be configured in time T to make product Y?

How can foundry X be modified so that it can make each product Y in time T?

How can foundry X be modified so that it can make product Y for cost C?

How can foundry X be modified so that it can be configured for cost C to make product Y?

How can foundry X be modified so that each product Y can be made for cost C?

How can foundry X be modified so that the overhead to make product Y has for cost C?

How can foundry X be modified to make product Y at rate R?

5.2 INTERMEDIATE LEVEL QUESTIONS

5.2.1   Questions During Configuration

Should foundry X make or buy subsystem S of product Y?

How should foundry X decompose product Y into subsystems S and T?

How should foundry X assemble subsystems S and T into product Y?

5.2.2   Qualitative Subsystem Questions

Above with cost/schedule qualities

5.2.3   Item Questions

Can foundry X add feature F to item U to yield item V?

What feature F should foundry X add to item U to yield item V?

Can foundry X use resource R to add feature F to item U?

What resource should foundry X use to add feature F to item U?

What process P and resource R should foundry X use to add feature F to item U?

Can foundry X assemble item U and item V into assembly A?

What process P and resource R should foundry X use to assemble item U and item V into assembly A?

Can foundry X assemble item U with subassembly S to produce assembly A?

What process P and resource R should foundry X use to assemble item U with subassembly S to produce assembly A?

Can foundry X move item U from station J to station K?

Can foundry X move item U from station J to station K with resource R and process P?

How can foundry X move item U from station J to station K?

5.2.4   Qualitative Item Questions

Above with cost/schedule qualities

5.3 LOWEST LEVEL QUESTIONS

5.3.1   Fabrication Questions

Can foundry X use Machine M and process P to add feature F to item U to yield item V?

How much does it cost foundry X to use Machine M and process P to add feature F to item U to yield item V?

How much does it cost for foundry X to configure Machine M to use process P to add feature F to item U to yield item V?

How long does it take for foundry X to use Machine M and process P to add feature F to item U to yield item V?

How long does it take for foundry X to configure Machine M to use process P to add feature F to item U to yield item V?

Can foundry X use Machine M, Tool T and process P to add feature F to item U to yield item V?

How much does it cost foundry X to use Machine M, Tool T and process P to add feature F to item U to yield item V?

How much does it cost for foundry X to configure Machine M to use Tool T and process P to add feature F to item U to yield item V?

How long does it take for foundry X to use Machine M, Tool T and process P to add feature F to item U to yield item V?

How long does it take for foundry X to configure Machine M to use Tool T and process P to add feature F to item U to yield item V?

5.3.2 Assembly Questions

Can foundry X use resource R and process P to assemble item U and item V into assembly A?

How much does it cost foundry X to use resource R and process P to assemble item U and item V into assembly A?

How much does it cost for foundry X to configure resource R and process P to assemble item U and item V into assembly A?

How long does it take for foundry X to use resource R and process P to assemble item U and item V into assembly A?

How long does it take for foundry X to configure resource R and process P to assemble item U and item V into assembly A?

Can foundry X move item U from station J to station K with resource R and process P?

How much does it cost foundry X to move item U from station J to station K with resource R and process P?

How much does it cost to configure foundry X to move item U from station J to station K with resource R and process P?

How long does it take for foundry X to move item U from station J to station K with resource R and process P?

How long does it take to configure foundry X to move item U from station J to station K with resource R and process P?

6.0 Appendix 4 – Document History

6.1 VERSION 13, 26 MARCH 2012

Added classification of services as either pure data or business logic. Added Document History appendix.

6.2 VERSION 14, 9 MAY 2012

Added sample SOAP messages to all available and limited availability services for which such sample messages were missing. Added entries for services 1.1.12, 1.1.17, 1.1.18, 1.1.19, 1.1.20, 1.1.21, 1.1.22, 1.3.14, 1.4.3, 1.4.5, 1.4.6, 1.4.7, and 1.4.8.

APPENDIX F – HAMML Synopsis

The Human-Assisted-Manufacturing Model Library (HAMML) for DARPA's iFAB program is a node of the Manufacturing Capability and Process Model Library (MCPML) wherein the key elements of human actor manufacturing activities and assembly processes are modeled. HAMML provides a human assembly process model that is integral to manufacturing process analysis for foundry configuration in determining process plans that satisfy workforce considerations and provide cost estimates of labor based on activity completion times. The manufacturing process models are developed for a data-driven, reconfigurable manufacturing facility. Relevant machine and tool resources are characterized for manufacturing processes to provide a broad characterization of manufacturing process coverage in a hypothetical final assembly factory for infantry fighting vehicles. A sample population of manufacturing process resources is provided with the library.

HAMML defines a library with data about human activities, machines, tools, processes, resources, aggregated process descriptions, and more. This library is presented to external tools via an interface that contains the logic and heuristics needed to place the manufacturing elements and resources into appropriate context for manufacturing processes. The interface provides data to queries for manufacturability, process sequencing, and foundry configuration. The service-oriented nature of the library interface makes it ready for integration with other AVM program tools and services as well as other performer MCPML repositories.

A prototype Virtual Manufacturing Environment (VME) was created to ascertain manufacturability from a crowd-sourced perspective and virtual configuration of the iFAB foundry. VME uses an open-source "virtual worlds" engine called Tundra. Assembly process information draws from the MCPML interface as XML data which shows, through use of avatars, manufacturing feedback on reachability, spatial configuration, part and work cell layout, and potential for collaboration between multiple users.

Microsoft SQL Server 2008 used as the database for the HAMML library. Raw source data for manufacturing process resources and activities are stored in Microsoft Excel Workbooks. The web-service-based interface to HAMML is developed in Microsoft Visual Studio and directly queries the SQL server database.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ANSI   American National Standards Institute

API   Application Programming Interface

ARL   Applied Research Laboratory

ASME   American Society of Mechanical Engineers

AVM   Adaptive Vehicle Make

BOM   Bill of Materials

C2M2L   Component, Context, and Manufacturing Model Library

CAD   Computer Aided Design

CICMHE   College Industry Council on Material Handling Education

CMU   Carnegie-Mellon University

CNC   Computer Numerically Controlled

COTS   Commercial Off-the-Shelf

DARPA   Defense Advanced Research Projects Agency

DDM   Direct Digital Manufacturing

GD&T   Geometrical Dimensions and Tolerances

GM   General Motors

GME   Generic Modeling Environment

GUID   Globally Unique Identifier

HAMML   Human-Assisted-Manufacturing Model Library

HVAC   Heating Ventilation Air Conditioning

IE   Industrial Engineering

iFAB   Instant Foundry Adaptive through Bits

ISO   International Standards Organization

IT   International Tolerance

IFV	Infantry Fighting Vehicle
MCPML	Manufacturing Capability and Process Model Library
MHE	Material Handling Equipment
MHIA	Material Handling Industry of America
MODAPTS	Modular Arrangement of Predetermined Time Standards
MRR	Material Removal Rate
MS&T	Missouri University of Science and Technology
MSD	Musculoskeletal Disorder
N	Newtons
NIST	National Institute of Standards and Technology
NOAA	National Oceanic and Atmospheric Administration
PI	Principal Investigator
QFD	Quality Function Deployment
SME	Subject Matter Expert
SOAP	Simple Object Access Protocol
TA	Technical Area
TARDEC	Tank Automotive Research, Development, and Engineering Center
TDP	Technical Data Package
VLSI	Very Large Scale Integration
VME	Virtual Manufacturing Environment
WSDL	Web Services Description Language
XML	Extensible Markup Language